

Centaur Intelligence as Topos and Algebra: A Unified Framework for Human–AI Reasoning, Gluing, and Learning

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

September 30, 2025

This paper proposes a precise mathematical framework for “centaur intelligence” that couples a human and a language model into a single, analyzable system. We develop two complementary formalisms. First, an algebraic entity—the Centaur Algebra—captures operations at the instance level: human and model convex state spaces, monoid actions for utterances and tokens, and affine couplings for prompt encoding, attention, prediction, and human update. This yields concrete invariants, morphisms, products, and quotients, supporting proofs about training, ensembling, and behavioral equivalence. Second, we construct a genuine topos of interactions using presheaves (or sheaves) over alternating human–model traces. The topos provides internal logic (subobject classifier), local-to-global reasoning via sheaf gluing (Kolmogorov-style consistency), and modalities via Lawvere–Tierney topologies for verification, safety, or observability. We then show how a concrete centaur instance induces a slice topos “over us,” while a geometric theory of centaur algebras admits a classifying topos whose points are exactly such systems. The result is a clean separation of concerns: algebra for *operations of one centaur*, topos for *logic over families, partial information, and gluing*. Together they support principled design, auditing, and governance of human–AI collaboration, turning slogans about hybrid intelligence into theorems about composition, consistency, and control.

Keywords: centaur intelligence, human–AI collaboration, convex algebra, monoid action, presheaf topos, sheaf gluing, Lawvere–Tierney topology, classifying topos, slice topos, probabilistic coalgebra, Markov categories, behavioral equivalence, safety modalities, verification, compositionality.

A collaboration with GPT-5. CC4.0. 71 pages.

Outline

1. **Motivation and Contributions**
 - 1.1 Why centaur intelligence needs math beyond metaphors
 - 1.2 Instance vs. ambient logic: algebra and topos as complementary tools
 - 1.3 Summary of results and guide to the paper
2. **Preliminaries (Copy-Safe)**
 - 2.1 Barycentric (convex) algebras and affine maps
 - 2.2 Free monoids Σ^* and U^* , actions, homomorphisms
 - 2.3 Presheaf and sheaf topoi over small sites; subobject classifier; internal logic
 - 2.4 Finitary distribution monad (objectwise) and coalgebraic notation
3. **Centaur Algebra: The Instance-Level Entity**
 - 3.1 Data: $(H, M, U_u, A_t, \text{enc}, \text{att}, \text{abs}, p, q, h_0, m_0)$
 - 3.2 One-step semantics and interleavings (step, mixed updates)
 - 3.3 Morphisms, subalgebras, products, quotients (bisimulation congruences)
 - 3.4 Forgetful functors and “free centaur” construction (sketch)
4. **The Centaur Topos: Contexts, Covers, and Gluing**
 - 4.1 Site Ctx : finite alternating traces and refinements
 - 4.2 Presheaf topos $E = \text{Set}^{(\text{Ctx}^{\text{op}})}$ vs. sheaf topos $\text{Sh}(\text{Ctx}, J)$
 - 4.3 Internal centaur algebra objectwise; naturality constraints
 - 4.4 Sheaf gluing as Kolmogorov consistency for local posteriors
5. **Modalities, Predicates, and Verification**
 - 5.1 Subobjects as predicates on runs and states
 - 5.2 Lawvere–Tierney topologies for verified vs. speculative knowledge
 - 5.3 Safety modalities, partial correctness, and closure/density properties
 - 5.4 From local certificates to global guarantees via gluing
6. **Universal Views: Slice and Classifying Topoi**
 - 6.1 Our concrete centaur as Spec : the slice topos $E_{\text{me}} = E / \text{Spec}$
 - 6.2 Geometric theory T_{cent} and its classifying topos E_{class}
 - 6.3 Points of E_{class} as centaur algebras; completeness and transport of structure
7. **Compositionality and Interfaces**
 - 7.1 Coupling multiple centaurs; products and pullbacks of interfaces
 - 7.2 Interoperation with external tools (retrievers, verifiers) as morphisms
 - 7.3 Diagrammatic laws in Markov-style categories (channel view)

8. Case Studies (Small but Realistic)

- 8.1 Two-turn assistant loop: observable predicates and gluing
- 8.2 Safety-gate as a modality; abstention and escalation
- 8.3 Ensemble (mix) vs. switch (action) and preserved properties

9. Learning, Adaptation, and Identifiability

- 9.1 Training as algebraic endomorphisms; invariants under fine-tuning
- 9.2 Behavioral equivalence, kernels of homomorphisms, and quotients
- 9.3 Limits of identifiability: what cannot be recovered from traces alone

10. Auditing, Governance, and Ethics as Modal Logic

- 10.1 Policy predicates, evidence operators, and update rules
- 10.2 From local audit findings to global claims by sheaf conditions
- 10.3 Fail-safe design patterns in the internal logic

11. Related Work and Bridges

- 11.1 Probabilistic automata, coalgebra, effect modules
- 11.2 Markov categories, quasi-toposes, sheaves of models
- 11.3 Program logics, epistemic modal logic, interactive theorem proving

12. Limitations and Future Directions

- 12.1 Expressiveness vs. tractability; when sheafification matters
- 12.2 Toward resource-aware or computationally graded variants
- 12.3 Empirical tooling: checkable axioms, canonical datasets, and benchmarks

Epilogue

Appendix A (Copy-Safe Axioms)

- A.1 Convexity axioms (mix)
- A.2 Monoid action axioms ($U_e = \text{id}$, $U_{vu} = U_u \circ U_v$; $A_e = \text{id}$, $A_{vu} = A_u \circ A_v$)
- A.3 Affineness of enc , att , abs , p , q
- A.4 Small-step semantics and coalgebraic presentation

Appendix B (Geometric Theory T_{cent})

- B.1 Signature, geometric sequents, and admissible rules
- B.2 Universal property of E_{class} and points-as-models

Appendix C (Worked Examples)

- C.1 Two-turn trace calculations (copy-safe)
- C.2 Minimal code sketch for objectwise distributions and gluing checks

References, Art

1. Motivation and Contributions

1.1 Why centaur intelligence needs math beyond metaphors

“Centaur intelligence” is often described in slogans: human judgment plus machine scale. That framing helps sell the idea but obscures what must be *proved* for safety, capability, and governance. Without crisp structure we cannot ask: Which properties compose under iteration? When do local checks imply global guarantees? What *invariants* survive fine-tuning or ensembling? We argue that a hybrid system needs (i) **instance-level operations**—how human and model states evolve, couple, and read out—and (ii) an **ambient logic**—how partial observations, predicates, and certifications glue into whole-system claims. Purely statistical or purely symbolic views each miss half the picture. By articulating a minimal algebra for the instance and a topos for the ambient logic, we replace metaphors with objects that admit homomorphisms, limits, subobjects, and modal operators—i.e., tools that support theorems rather than anecdotes.

1.2 Instance vs. ambient logic: algebra and topos as complementary tools

Our split is deliberate:

- **Algebra (instance):** A Centaur Algebra $\mathcal{C} = (H, M, U_{(-)}, A_{(-)}, enc, att, abs, p, q, h0, m0)$ specifies *operations* on convex state spaces and the monoid actions that drive them. It supports morphisms, products, subalgebras, and quotients, enabling reasoning about training, ensembling (mix), and behavioral equivalence (bisimulation-style quotients).
- **Topos (ambient):** A presheaf/sheaf topos over alternating human–model traces provides the *logic of partial information*: predicates as subobjects, sheaf gluing for local-to-global consistency (Kolmogorov-style), and Lawvere–Tierney topologies to model verification, safety, or observability modalities. Slicing at a concrete instance yields a “universe over us”; classifying the geometric theory yields a universal space of all centaur models.

These roles are orthogonal and synergistic: algebra proves *what the system does*; the topos proves *what we can know and certify about it*, locally and then globally.

1.3 Summary of results and guide to the paper

- **Definition (Centaur Algebra):** We introduce a minimal, copy-safe signature capturing human/model convexity, dual monoid actions, and affine couplings. We show the category $\text{Cent}(\text{Sigma}, U)$ has standard constructions (products, subalgebras, quotienting by behavioral kernels) and useful forgetful/free functors to the human-only and model-only fragments.

- **Construction (Centaur Topos):** We define a site of finite alternating traces and build a presheaf (or sheaf) topos in which a centaur instance internalizes objectwise. Gluing theorems coincide with Kolmogorov consistency across overlapping windows.
- **Modalities for assurance:** We exhibit Lawvere–Tierney topologies that encode verification modes (e.g., “checked,” “speculative”), yielding modal logics for safety and audit that compose along covers.
- **Universal viewpoints:** (a) **Slice topos over us** for reasoning relative to a fixed system; (b) a **classifying topos** for the geometric theory T_{cent} , whose points are exactly centaur algebras.
- **Case studies:** Small, realistic loops show how predicates (safety gates, abstention, escalation) become subobjects and how local certificates glue.
- **Limits and outlook:** We chart identifiability boundaries from traces, when sheafification changes conclusions, and how to add resource-aware gradings.

Reading guide. Section 2 fixes preliminaries. Section 3 develops the algebra (operations, morphisms, invariants). Section 4 builds the topos of interactions and proves gluing lemmas. Section 5 introduces modalities for verification/safety. Section 6 presents slice/classifying constructions. Sections 7–9 treat composition, case studies, and learning/identifiability. Section 10 frames auditing and governance as modal logic. Section 11 situates related work; Section 12 covers limitations and future directions. Appendices give copy-safe axioms, the geometric theory, and worked examples.

2. Preliminaries (Copy-Safe)

2.1 Barycentric (convex) algebras and affine maps

Definition (barycentric/convex algebra).

A set H with binary operations $\text{mix}_{\lambda} : H \times H \rightarrow H$ for each rational λ in $[0,1]$ is a barycentric algebra if the following axioms hold for all x,y,z in H :

$$(B1) \quad \text{mix}_1(x,y) = x$$

$$(B2) \quad \text{mix}_{\lambda}(x,y) = \text{mix}_{\{1-\lambda\}}(y,x)$$

$$(B3) \quad \text{mix}_{\lambda}(\text{mix}_{\mu}(x,y), z)$$

$$= \text{mix}_{\{\lambda*\mu\}}(x, \text{mix}_{\{\lambda*(1-\mu)/(1-\lambda*\mu)\}}(y, z))$$

whenever $\lambda*\mu \neq 0$ and $\lambda*\mu \neq 1$

A more presentation-friendly axiom system (equivalent on finitary convex combinations) uses a nullary constant-family operator that builds finite convex sums $\sum_i \lambda_i * x_i$ with $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. We adopt the binary schema above for copy-safe simplicity.

Examples.

1. Real vector spaces with $\text{mix}_{\lambda}(x,y) = \lambda * x + (1-\lambda) * y$.
2. Probability simplices $\text{Dist}_{\text{fin}}(Z)$ (finitely supported distributions on Z) with convex combination of measures.
3. Subsets of a vector space closed under convex combination.

Affine maps.

A function $f : H \rightarrow K$ between barycentric algebras is affine if $f(\text{mix}_{\lambda}(x,y)) = \text{mix}_{\lambda}(f(x), f(y))$ for all x,y and all rationals λ in $[0,1]$. Affineness extends to all finite convex sums.

2.2 Free monoids Σ^* and U^* , actions, homomorphisms

Free monoid.

Given a set (alphabet) Σ , the free monoid Σ^* is the set of all finite words over Σ (including the empty word ϵ), with operation given by concatenation and unit ϵ . Similarly define U^* for a set U (human actions).

Monoid action.

Let M be a set. A (right) action of Σ^* on M is a function $\text{act} : M \times \Sigma^* \rightarrow M$ such that:

$$(A1) \text{act}(m, \epsilon) = m$$

$$(A2) \text{act}(m, vw) = \text{act}(\text{act}(m, v), w)$$

For readability we write $A_w(m)$ for $\text{act}(m,w)$, and for generators t in Σ write $A_t : M \rightarrow M$. An action is **by endomorphisms of structure** when each A_t preserves the designated operations on M (e.g., is affine on a convex algebra).

Monoid homomorphism.

A map $\phi : \Sigma^* \rightarrow T$ into a monoid $(T, *, e)$ is a homomorphism if

$$\phi(\epsilon) = e$$

$$\phi(vw) = \phi(v) * \phi(w)$$

In particular, a monoid action is the same as a homomorphism $\rho : \Sigma^* \rightarrow \text{End}(M)$ into the endomorphism monoid of M .

Product actions and interleavings.

Given actions U^* on H and Σ^* on M , the product set $H \times M$ carries the independent action $(u, w) \cdot (h, m) := (U_u(h), A_w(m))$. Interleavings over alternating traces are handled at the site level (Sec. 2.3).

2.3 Presheaf and sheaf topoi over small sites; subobject classifier; internal logic

Small site of contexts.

Let Ctx be a small category whose objects are finite alternating traces

$w = (u_0, t_1, u_1, t_2, \dots, u_k, t_k)$ with u_i in U , t_j in Σ

and whose morphisms are prefix embeddings, restrictions, or refinement maps between traces.

Presheaf topos.

$\text{PSh}(\text{Ctx}) := \text{Set}^{\text{Ctx}^{\text{op}}}$ is the category of presheaves on Ctx (contravariant functors to Set) with natural transformations as morphisms. This is a Grothendieck topos. It has:

- all finite limits and colimits,
- exponentials B^A (objectwise ends),
- a subobject classifier Ω (sieves objectwise).

Grothendieck topology and sheaves.

Choose a coverage J on Ctx (e.g., a family $\{u_i \rightarrow w\}$ is covering iff the u_i jointly refine/cover w in time). The category $\text{Sh}(\text{Ctx}, J)$ of sheaves for J is a (Grothendieck) topos obtained from $\text{PSh}(\text{Ctx})$ by enforcing the sheaf condition:

Matching families on a J -cover have a unique amalgamation.

Internal logic.

Every Grothendieck topos supports higher-order intuitionistic logic:

- **Predicates** on an object X are subobjects $S \rightarrow X$.
- **Truth values** form Ω ; characteristic maps $\chi_S : X \rightarrow \Omega$ classify subobjects.
- **Quantifiers** arise from adjoints to pullback along maps; exponentials supply implication and higher-order function spaces.

- In $\text{Sh}(\text{Ctx}, J)$, local-to-global reasoning is done by the sheaf condition; “gluing” of compatible local data corresponds to existence/uniqueness of amalgamations.

Internalization pattern.

Algebraic data like convex operations, monoid actions, and affine maps are specified **objectwise** with naturality across restriction maps in Ctx . Thus a “centaur algebra” can be made an **internal object** of the topos.

2.4 Finitary distribution monad (objectwise) and coalgebraic notation

Finitary distributions.

For a set X , let $D_{\text{fin}}(X)$ denote the set of finitely supported probability measures on X :

$$D_{\text{fin}}(X) = \{ \mu : X \rightarrow \mathbb{R}_{\geq 0} \mid \text{supp}(\mu) \text{ finite, } \sum_{x \in X} \mu(x) = 1 \}.$$

Monad structure on Set.

- Unit $\eta_X : X \rightarrow D_{\text{fin}}(X)$ given by Dirac: $\eta_X(x) = \delta_x$.
- Multiplication $\mu_X : D_{\text{fin}}(D_{\text{fin}}(X)) \rightarrow D_{\text{fin}}(X)$ given by expectation (flattening):
for ν in $D_{\text{fin}}(D_{\text{fin}}(X))$, define $\mu_X(\nu)(x) = \sum_{p \in D_{\text{fin}}(X)} \nu(p) * p(x)$.

This makes $(D_{\text{fin}}, \eta, \mu)$ a monad on Set . Eilenberg–Moore algebras for D_{fin} are exactly **convex (barycentric) algebras** with affine structure maps; Kleisli arrows $X \rightarrow D_{\text{fin}}(Y)$ are **stochastic maps**.

Objectwise lift to a topos.

In a presheaf (or sheaf) topos $\mathcal{E}$ over Ctx , define D_{fin} **objectwise**:

$$(D_{\text{fin}} F)(w) := D_{\text{fin}}(F(w)), \quad (D_{\text{fin}} F)(r) := D_{\text{fin}}(F(r))$$

for each presheaf $F : \text{Ctx}^{\text{op}} \rightarrow \text{Set}$ and arrow $r : u \rightarrow w$ in Ctx . The monad laws hold objectwise, yielding a monad on $\mathcal{E}$. Internal convex algebras are Eilenberg–Moore algebras for this lifted monad.

Coalgebraic notation.

Given an endofunctor F (e.g., $F(X) = D_{\text{fin}}(\Sigma \times X)$), an F -coalgebra is a map

$$c : X \rightarrow F(X).$$

Intuitively, c gives one step of observable behavior (e.g., next-token distribution and next state). For centaur systems we often use a product functor

$$F(X) = D_{\text{fin}}(\Sigma) \times X \text{ or a richer}$$

$F(X) = D_{\text{fin}}(\text{Sigma} \times X)$ (joint emission and next-state law).

Coalgebra morphisms commute with c ; final coalgebras (when they exist) characterize observable behaviors up to bisimulation. In our setting we pair coalgebraic one-step views with the algebraic operations (convex structure, monoid actions) and interpret them **internally** in the chosen topos.

Takeaway.

- D_{fin} supplies probability;
- convex algebras are the Eilenberg–Moore side (averaging/affinity);
- coalgebras model stepwise dynamics;
- presheaf/sheaf topoi provide the ambient logic so that local probabilistic views glue into global ones.

3. Centaur Algebra: The Instance-Level Entity

3.1 Data: $(H, M, U_u, A_t, \text{enc}, \text{att}, \text{abs}, p, q, h_0, m_0)$

Carriers and basic structure.

- H, M are barycentric (convex) algebras with mix_{λ} operations (Sec. 2.1).
- U, Sigma are alphabets for human utterances and model tokens; U^*, Sigma^* are free monoids (Sec. 2.2).

Actions (affine).

- Human side: for each u in U , an affine map $U_u : H \rightarrow H$, extended to U_w for w in U^* with $U_e = \text{id}_H$ and $U_{\{vu\}} = U_u \circ U_v$.
- Model side: for each t in Sigma , an affine map $A_t : M \rightarrow M$, extended to A_w with $A_e = \text{id}_M$ and $A_{\{vt\}} = A_t \circ A_v$.

Couplings (affine).

- $\text{enc} : H \rightarrow \text{Msg}$ (encode human state to a message/prompt type; Msg is any chosen set).
- $\text{att} : H \times M \rightarrow M$ (model attends to human; often $\text{att}(h, m) = A_{\{\phi(\text{enc}(h))\}}(m)$ for some coding ϕ).
- $\text{abs} : H \times \text{Dist}_{\text{fin}}(\text{Sigma}) \rightarrow H$ (human absorbs a distribution over tokens).

Readouts (affine).

- $p : M \rightarrow \text{Dist\_fin}(\text{Sigma})$ (model next-token law).
- $q : H \rightarrow \text{Obs}$ (optional human externalization/observation type).

Initial point.

- $h_0 : 1 \rightarrow H, m_0 : 1 \rightarrow M$.

Centaur state.

$X := H \times M$ with product convex structure (componentwise mix). All structure maps are required to be affine in their convex arguments.

3.2 One-step semantics and interleavings (step, mixed updates)**One-step without explicit human action.**

Given (h, m) in X :

$m' := \text{att}(h, m)$

$y := p(m')$ in $\text{Dist\_fin}(\text{Sigma})$

$h' := \text{abs}(h, y)$

$\text{step}(h, m) := (h', m')$.

With explicit utterance u in U .

Apply the human action before coupling:

$h_1 := U_u(h)$

$m' := \text{att}(h_1, m)$

$y := p(m')$

$h' := \text{abs}(h_1, y)$

$\text{step}_u(h, m) := (h', m')$.

Token-conditioned model transition (optional).

If we expose a realized token $t \sim y$, one may define a stochastic small-step

$m'' := A_t(m')$

$h'' := \text{abs}(h_1, \text{delta}_t)$

and lift the above to a Kleisli arrow $X \rightarrow D_{\text{fin}}(X)$. Our default keeps p 's distribution at the readout and remains deterministic in state space while stochastic in outputs.

Interleavings over traces.

For an alternating trace

$w = (u_0, t_1, u_1, t_2, \dots, u_k, t_k),$

we compose the corresponding U- and A-actions with step/att/abs in the obvious order (copy-safe pseudocode):

$(h, m) := (h_0, m_0)$

for $j=0..k$:

$h := U_{\{u_j\}}(h)$

$m := \text{att}(h, m)$

$y := p(m)$

$h := \text{abs}(h, y)$

if $j < k$:

$m := A_{\{t_{j+1}\}}(m)$ # optional realized-token update

return $(h, m), y$ # final state and last distribution

Mixtures/ensembles.

For parallel centaurs C_i with weights λ_i ($\sum \lambda_i = 1$), define the product carrier

$H_{\text{ens}} := \prod_i H_i, \quad M_{\text{ens}} := \prod_i M_i,$

and the *mixed readout*

$p_{\text{ens}}((m_i)_i) := \sum_i \lambda_i * p_i(m_i).$

Affineness of att, abs ensures step_{ens} commutes with convex mixing at the readout.

3.3 Morphisms, subalgebras, products, quotients (bisimulation congruences)

Morphisms.

A morphism $f : C \rightarrow C'$ is a pair of affine maps $(f_H : H \rightarrow H', f_M : M \rightarrow M')$ such that:

$f_H \circ U_u = U'_u \circ f_H \quad \text{for all } u \text{ in } U$

$$f\_M \circ A\_t = A'_t \circ f\_M \quad \text{for all } t \text{ in } \Sigma$$

$$\text{enc}' \circ f\_H = \text{enc}$$

$$\text{att}' \circ (f\_H \times f\_M) = f\_M \circ \text{att}$$

$$\text{abs}' \circ (f\_H \times \text{id}) = f\_H \circ \text{abs} \quad (\text{where id is on } \text{Dist\_fin}(\Sigma))$$

$$p' \circ f\_M = p$$

$$q' \circ f\_H = q$$

$$f\_H(h_0) = h'_0, \quad f\_M(m_0) = m'_0.$$

(When enc/att/abs are factored differently, adjust the equalities accordingly; the core idea is commutation with structure.)

Subalgebras.

A sub-centaur C_S consists of convex-closed subsets $H\_S \leq H$, $M\_S \leq M$ such that

$$U\_u(H\_S) \text{ subseq } H\_S, \quad A\_t(M\_S) \text{ subseq } M\_S,$$

$$\text{att}(H\_S \times M\_S) \text{ subseq } M\_S, \quad \text{abs}(H\_S \times \text{Dist\_fin}(\Sigma)) \text{ subseq } H\_S,$$

$$h_0 \text{ in } H\_S, \quad m_0 \text{ in } M\_S.$$

Structure maps restrict, giving an induced centaur algebra.

Products.

Given centaurs C^1, C^2 over the same (U, Σ) , their product $C^1 \times C^2$ has carriers $H^1 \times H^2$ and $M^1 \times M^2$ with componentwise structure:

$$U\_u(h_1, h_2) := (U\_u^1(h_1), U\_u^2(h_2)), \quad A\_t(m_1, m_2) := (A\_t^1(m_1), A\_t^2(m_2)),$$

$$\text{att}((h_1, h_2), (m_1, m_2)) := (\text{att}^1(h_1, m_1), \text{att}^2(h_2, m_2)),$$

$$\text{abs}((h_1, h_2), y) := (\text{abs}^1(h_1, y), \text{abs}^2(h_2, y)),$$

$$p(m_1, m_2) := \text{mix\_lambda}(p^1(m_1), p^2(m_2)) \quad (\text{choose lambda if a convex sum is desired})$$

or keep p componentwise and introduce an external combiner; both variants are useful.

Quotients via behavioral kernels (bisimulation-style).

Let $\sim\_H$ be a congruence on H and $\sim\_M$ a congruence on M (equivalence relations closed under mix and respected by all structure maps). The quotient carriers $H/\sim\_H$ and $M/\sim\_M$ inherit well-defined structure if, for all related pairs, the following hold:

$$h \sim\_H h' \text{ and } m \sim\_M m' \Rightarrow \text{att}(h, m) \sim\_M \text{att}(h', m')$$

$h \sim_H h'$ and $y=y'$ $\Rightarrow$ $\text{abs}(h,y) \sim_H \text{abs}(h',y')$

$m \sim_M m'$ $\Rightarrow$ $p(m) = p(m')$ (or $p(m) \sim p(m')$ if coarser equality is intended)

Such quotients identify *behaviorally indistinguishable* states and produce smaller centaur algebras. Existence follows from standard quotienting of convex algebras under affine-congruence plus well-definedness of induced maps.

Bisimulation kernel of a morphism.

For a morphism $f : C \rightarrow C'$, define

$\sim_H := \ker(f_H)$ and $\sim_M := \ker(f_M)$.

These are congruences; the coequalizer $C \rightarrow C/\sim \rightarrow C'$ is universal among maps factoring through f .

3.4 Forgetful functors and “free centaur” construction (sketch)

Forgetful functors.

There are structure-preserving forgetful functors:

$U_H : \text{Cent}(\text{Sigma}, U) \rightarrow \text{HumanAlg}(U)$ (drop M-side, keep H , U_u , abs , q , h_0)

$U_M : \text{Cent}(\text{Sigma}, U) \rightarrow \text{LLMAlg}(\text{Sigma})$ (drop H-side, keep M , A_t , p , m_0)

$U_{\text{pair}} : \text{Cent}(\text{Sigma}, U) \rightarrow \text{ConvAlg} \times \text{ConvAlg}$ (drop actions/couplings)

Each creates/reflects limits that are computed componentwise (products, equalizers) provided mix, actions, and couplings are affine and objectwise defined.

Left adjoints (free constructions).

Sketch the left adjoint F_M to U_M (free centaur on a model-side algebra) and F_H to U_H :

- **Free on model (attach a trivial human):**
Given (M, A_t, p, m_0) , form $H := F(M)$ as the *free convex algebra* generated by a point h_0 and a copy of $\text{Dist}_{\text{fin}}(\text{Sigma})$; equip:
 - $U_u := \text{id}_H$ (no nontrivial human dynamics by default)
 - $\text{enc}(h) := e_0$ (fixed prompt symbol or projection from H if desired)
 - $\text{att}(h, m) := m$ (no extra attention without human effect)
 - $\text{abs}(h, y) := \text{combine}(h, y)$ (the universal affine extension that records y)
 - $q :=$ projection from H to an Obs generator

Universal property: any centaur $(H', M, \dots)$ over the same model receiving an affine map from $\text{Dist\_fin}(\text{Sigma})$ factors uniquely through H via an affine homomorphism preserving structure. Variants add a minimal nontrivial enc/att by freely adjoining a morphism $H \rightarrow \text{Msg}$ and its affine extension.

- **Free on human (attach a trivial model):**

Dually, given (H, U_u, q, h_0) , set $M := F'(H)$ as the free convex algebra on a point m_0 and a generator for Sigma ; let:

- $A_t := \text{id}_M$
- $p(m) := y_0$ (fixed baseline distribution)
- $\text{att}(h, m) := m$
- $\text{abs}(h, y) := U_{\{u(y)\}}(h)$ (optionally route distributions to scripted human updates)

Universal property is dual.

- **Free on a pair (H, M) (no actions given):**

Adjoin actions U_u and A_t freely as endomorphisms respecting convexity, then adjoin enc, att, abs, p, q freely as affine maps subject only to typing and optional side-axioms (e.g., $p \circ A_t$ affine, att affine). This produces the *initial centaur* mapping to any concrete centaur extending those generators.

Remark on practicality.

In applications, one typically uses *semi-free* constructions: start from given (H, M) and chosen lightweight axioms for enc/att/abs (e.g., $\text{att}(h, m) = A_{\{\phi(\text{enc}(h))\}}(m)$ with ϕ linear), then quotient by a behavioral kernel to enforce identifications discovered empirically.

Takeaway.

- Morphisms formalize “compatibility-preserving adapters” between centaurs.
- Subalgebras/quotients give restriction and minimization (bisimulation).
- Products and (semi-)free constructions support modular design and composition.
- All of this is instance-level; Sec. 4 will lift these structures objectwise into a presheaf/sheaf topos, enabling local-to-global reasoning and modalities.

4. The Centaur Topos: Contexts, Covers, and Gluing

4.1 Site Ctx: finite alternating traces and refinements

Objects (contexts).

A context is a finite alternating trace

$w = (u_0, t_1, u_1, t_2, \dots, u_k, t_k)$ with u_i in U , t_j in Σ , $k \geq 0$.

Intuition: u 's are human utterances/edits, t 's are model tokens/actions. We allow $k=0$ (pure-human prefix).

Morphisms (refinements/restrictions).

A morphism $r : v \rightarrow w$ witnesses that v is a *refinement* (or restriction) of w . We use three generators, closed under composition:

1. **Prefix embedding:** drop a tail, $\text{pref} : (u_0, \dots, u_k, t_k) \rightarrow (u_0, \dots, u_j, t_j)$ for $j \leq k$.
2. **Window restriction:** extract a contiguous subtrace $\text{slice}_{[i..j]} : w \rightarrow w[i..j]$.
3. **Token coarsening/refinement:** optionally refine a token step into micro-steps or coarsen micro-steps back into one (used if we model realized-token transitions).

All arrows are deterministic structure maps describing *how to view a big trace locally*. The category is small by fixing finite alphabets or by bounding length.

Covers.

A family $\{r_i : v_i \rightarrow w\}$ is a J -cover if the v_i jointly “see” every position of w and overlap along coherent subwindows. Concretely, we choose one (simple, useful) basis:

$J_{\text{cov}}(w) := \{\text{partitions of } w \text{ into overlapping windows of length } \leq L\}$

whose union is all of w and whose pairwise overlaps are windows too $\}$.

Parameter L (cover radius) is design- and theorem-dependent.

4.2 Presheaf topos $E = \text{Set}^{(\text{Ctx}^{\text{op}})}$ vs. sheaf topos $\text{Sh}(\text{Ctx}, J)$

Presheaves.

A presheaf $F : \text{Ctx}^{\text{op}} \rightarrow \text{Set}$ assigns:

- to each w : a set $F(w)$ (sections over w),
- to each $r : v \rightarrow w$: a restriction map $F(r) : F(w) \rightarrow F(v)$,

with functorial laws (identities, composition).

Topos structure of E.

$E = \text{Set}^{(\text{Ctx}^{\text{op}})}$ is a Grothendieck topos:

- finite limits/colimits computed objectwise,
- exponentials B^A exist,
- subobject classifier Ω is the sieve presheaf:
 $\Omega(w) = \{ \text{sieves } S \text{ on } w \}$ and $\Omega(r)$ is pullback of sieves.

Sheaves.

Given a Grothendieck topology J (Sec. 4.1), a presheaf F is a **sheaf** if, for every cover $\{ r_i : v_i \rightarrow w \}$ and every compatible family $(s_i \text{ in } F(v_i))$ with

$$F(p_{ij})(s_i) = F(p_{ji})(s_j) \text{ on } v_i \times_w v_j,$$

there exists a unique s in $F(w)$ such that $F(r_i)(s) = s_i$ for all i . The full subcategory of sheaves is a topos $\text{Sh}(\text{Ctx}, J)$. Intuition: *locally coherent data glue to a unique global section*.

When to use which.

- E (presheaves) for maximal generality and simple constructions.
- $\text{Sh}(\text{Ctx}, J)$ (sheaves) when local-to-global certainty is required (e.g., consistency of marginals implies existence of a global law).

4.3 Internal centaur algebra objectwise; naturality constraints

We internalize the Centaur Algebra (Sec. 3) as presheaves with structure maps that are **natural** in w .

Objects.

$H, M : \text{Ctx}^{\text{op}} \rightarrow \text{Set}$ (human/model state presheaves)

$\text{Msg}, \text{Obs} : \text{Ctx}^{\text{op}} \rightarrow \text{Set}$ (message/observation types; often constant)

Structure (all natural in w).

- Human actions: for each u in U ,
- $U_u(w) : H(w) \rightarrow H(w)$

s.t. $H(r) \circ U_u(w) = U_u(v) \circ H(r)$ for every $r : v \rightarrow w$. The U_* satisfy monoid axioms objectwise.

- Model actions: for each t in Σ ,

- $A_t(w) : M(w) \rightarrow M(w)$

natural in w , satisfying monoid axioms objectwise.

- Couplings:
- $enc(w) : H(w) \rightarrow \text{Msg}(w)$
- $att(w) : H(w) \times M(w) \rightarrow M(w)$
- $abs(w) : H(w) \times D_{fin}(\Sigma) \rightarrow H(w)$

each natural in w (commuting with restriction). Affine in convex arguments objectwise.

- Readouts:
- $p(w) : M(w) \rightarrow D_{fin}(\Sigma)$
- $q(w) : H(w) \rightarrow \text{Obs}(w)$

natural and affine objectwise.

- Points:
- $h_0 : 1 \rightarrow H, m_0 : 1 \rightarrow M$

i.e., chosen global sections (compatible under all restrictions).

Convexity internal to the topos.

D_{fin} lifts objectwise (Sec. 2.4). Declaring H, M as **Eilenberg–Moore algebras** for D_{fin} objectwise makes each $H(w), M(w)$ convex; naturality of the structure maps ensures the convex operations commute with restriction.

Naturality schema (copy-safe). For every $r : v \rightarrow w$:

$$H(r) \circ U_u(w) = U_u(v) \circ H(r)$$

$$M(r) \circ A_t(w) = A_t(v) \circ M(r)$$

$$M(r) \circ att(w) = att(v) \circ (H(r) \times M(r))$$

$$H(r) \circ abs(w) = abs(v) \circ (H(r) \times id_{\{D_{fin}(\Sigma)\}})$$

$$p(v) \circ M(r) = p(w)$$

$$q(v) \circ H(r) = q(w)$$

These equalities guarantee that semantics do not depend on which window we use to view a run.

4.4 Sheaf gluing as Kolmogorov consistency for local posteriors

Posterior presheaf.

Given the internal centaur, define a presheaf of *local posteriors*:

$$\text{Post}(w) := \{ y_w : M(w) \rightarrow D_{\text{fin}}(\Sigma) \mid y_w \text{ is } p(w) \text{ after } \text{att}(w) \text{ with a fixed } h \text{ in } H(w) \}$$

More concretely, fix a local human section h in $H(w)$; the induced posterior is

$$y_w(m) := p(w)(\text{att}(w)(h, m)).$$

When we restrict along $r : v \rightarrow w$, we map h to $H(r)(h)$ and m to $M(r)(m)$, hence:

$$\text{Post}(r)(y_w) := y_v \text{ where } y_v(m_v) = p(v)(\text{att}(v)(H(r)(h), m_v)).$$

Kolmogorov-style consistency.

Let $\{r_i : v_i \rightarrow w\}$ be a J -cover. A family of local posteriors (y_i in $\text{Post}(v_i)$) is **consistent on overlaps** if for every overlap square

$$v_{ij} = v_i \times_w v_j$$

$$p_{ij} : v_{ij} \rightarrow v_i, \quad p_{ji} : v_{ij} \rightarrow v_j$$

we have

$$\text{Post}(p_{ij})(y_i) = \text{Post}(p_{ji})(y_j) \text{ in } \text{Post}(v_{ij}).$$

Intuition: the marginals agree on all shared subwindows.

Gluing theorem (sheaf view).

If Post is a **sheaf** for J , then every J -compatible family (y_i) over $\{v_i \rightarrow w\}$ has a **unique** amalgamation y in $\text{Post}(w)$ with $\text{Post}(r_i)(y) = y_i$. This is exactly the Kolmogorov property: *agreeing local posteriors determine a unique global posterior on the larger window.*

When is Post a sheaf?

Sufficient (copy-safe) conditions:

1. **Naturality:** all structure maps for att , p , H , M commute with restriction (Sec. 4.3).
2. **Stability under overlaps:** att preserves pullbacks of windows:
3. $\text{att}(v_{ij}) = \text{att}(v_i)$ pulled back along $(H(p_{ij}), M(p_{ij}))$

and similarly for p .

4. **Determinacy of amalgamation:** for any span of windows forming a cover, the value of y_w on each m in $M(w)$ is determined by values on restrictions $M(r_i)(m)$ and these are

compatible.

Under these, Post satisfies the sheaf axiom for the cover basis J_{cov} .

Practical reading.

- Local evaluations of next-token laws (conditioned on local human state) that agree on every overlap glue to a unique law on the larger interaction window.
- This lets you prove *from local audits to global guarantees*: if safety predicates (subobjects of Post) are stable under restriction and jointly cover the session, the global posterior inherits safety.

Remark (distributions on projections).

The same pattern applies to *state* presheaves when you expose realized tokens. Define a Kleisli presheaf

$\text{Step}(w) : H(w) \times M(w) \rightarrow D_{\text{fin}}(H(w) \times M(w))$

from Sec. 3.2. Consistency of local transition laws on overlaps (marginal agreement) yields a glued global transition law, i.e., a well-defined one-step coalgebra over the larger window.

Takeaway.

- The site encodes “where” and “how finely” we observe centaur interactions.
- The topos supplies the subobject classifier and internal logic.
- The sheaf condition turns Kolmogorov-style marginal consistency into a *theorem of gluing*: local posteriors (and safety predicates about them) uniquely determine global ones.

5. Modalities, Predicates, and Verification

5.1 Subobjects as predicates on runs and states

Predicates = subobjects. In a Grothendieck topos E , a *predicate* on an object X is a monomorphism $S \rightarrow X$. By the subobject classifier Ω , every subobject is classified by a map $\chi_S : X \rightarrow \Omega$.

Typical choices in our setting:

- Predicates on *states*: $\text{Safe}_H \leq H$, $\text{Safe}_M \leq M$, $\text{Guard}_X \leq X = H \times M$.
- Predicates on *posteriors*: $\text{GoodPost} \leq \text{Post}$ (Sec. 4.4).
- Predicates on *runs/windows*: $\text{Admissible} \leq \text{Runs}(w)$ (if we expose realized traces).

Boolean operations. Intersections, unions, and implications are computed via pullback, image, and exponentials using Omega's Heyting structure. For a map $f : X \rightarrow Y$, *reindexing* a predicate is pullback: $f^*(T) := X \times_Y T$.

Hoare triples (internal view). Given a (possibly internal) transition step $X \rightarrow X$ or Kleisli arrow $X \rightarrow D_{\text{fin}}(X)$, a *partial-correctness sequent* is:

$$\{P\} \text{ step } \{Q\} ::= P \leq \text{step}^*(Q)$$

where $P, Q \leq X$ and step^* pulls predicates back along step (Kleisli case uses expectation/healthiness conditions; see 5.3).

5.2 Lawvere–Tierney topologies for verified vs. speculative knowledge

A **Lawvere–Tierney (L–T) topology** on E is an endomorphism $j : \Omega \rightarrow \Omega$ satisfying

- (i) $\text{true} \leq j(\text{true})$
- (ii) $j(p) \leq j(j(p))$
- (iii) $j(p \text{ and } q) = j(p) \text{ and } j(q)$ (preserves finite meets)

Intuitively, j upgrades a truth value to "verified truth." It induces:

- **j-closure** of a subobject $S \leq X$: the smallest j -closed subobject containing S .
- **j-dense** monos: $m : S \rightarrow X$ is j -dense iff its j -closure is all of X .
- **Sheafification for j** : the full subcategory of j -sheaves (objects orthogonal to j -dense monos) is a reflective subtopos. The reflector is the "verify" modality.

Two canonical j 's for centaur auditing.

1. **Evidence modality j_{evd} .** Interprets "supported by evidence": at each window w , j_{evd} maps a sieve to the sieve generated by covers along which there exists a *certificate section* in a fixed evidence presheaf Evd . Operationally, j_{evd} closes a predicate under *verifiable refinement*.
2. **Robustness modality j_{rob} .** Interprets "stable under admissible perturbations": choose a coverage of perturbation maps $\pi_i : v \rightarrow w$ (paraphrase edits, harmless prompt reorderings). Define j_{rob} to keep only those sieves stable under all π_i . Predicates closed under j_{rob} are *robust claims*.

Both give $\text{Sh}_j(E)$ (the j -sheaf subtopos) where reasoning is "as if verified/robust."

5.3 Safety modalities, partial correctness, and closure/density properties

Safety as a modality. Let $\text{Safe} \leq X$ be a predicate on centaur states (or on Post). Define the *safety modality* $\text{box\_Safe} : \text{Sub}(X) \rightarrow \text{Sub}(X)$ by

$\text{box\_Safe}(Q) := \text{largest } R \leq X \text{ such that } R \text{ and } \text{Safe} \leq Q.$

Internally this is intersection with Safe followed by j -closure if we require verification:

$\text{box\_Safe}^j(Q) := j\text{-closure}(\text{Safe and } Q).$

Dually, a *liveness* or *possibility* operator can be modeled via density or by left Kan extensions in presheaves; operationally we use

$\text{diamond\_Safe}(Q) := j\text{-dense-closure}(\text{Safe or } Q),$

to reflect "can reach Q while maintaining Safe on the path" in a one-step setting (extend to paths via induction or coalgebra).

Partial correctness under stochastic step. For a Kleisli arrow $\text{Step} : X \rightarrow D_{\text{fin}}(X)$ we use *healthiness* conditions (expectation-style):

- A predicate transformer $\text{wp\_Step} : \text{Sub}(X) \rightarrow \text{Sub}(X)$ (weakest precondition) defined by

$\text{wp\_Step}(Q) := \{x \text{ in } X \mid E_{\{x' \sim \text{Step}(x)\}}[1_Q(x')] = 1\}.$

This is the largest predicate P with $\{P\} \text{Step } \{Q\}$. If we work under j , use

$\text{wp\_Step}^j(Q) := j\text{-closure}(\text{wp\_Step}(Q))$

so proofs live in the verified subtopos.

Closure/density lemmas (copy-safe).

- *j-closure monotonicity:* $P \leq Q$ implies $j\text{-closure}(P) \leq j\text{-closure}(Q)$.
- *box_Safe^j is monotone and idempotent:*
 $Q \leq Q' \Rightarrow \text{box\_Safe}^j(Q) \leq \text{box\_Safe}^j(Q')$, and $\text{box\_Safe}^j(\text{box\_Safe}^j(Q)) = \text{box\_Safe}^j(Q)$.
- *Safety preservation under affine maps:* If $f : X \rightarrow X'$ is affine/natural and $\text{Safe}' \leq X'$, then $f^*(\text{Safe}')$ is convex-closed; if f respects j , then $j\text{-closure}(f^*(S)) = f^*(j\text{-closure}(S))$.

These give compositional rules for verified safety claims.

5.4 From local certificates to global guarantees via gluing

Local certificates. Suppose we have a cover $\{r_i : v_i \rightarrow w\}$ in Ctx and, for each i , a *certificate section* c_i proving $\{P_i\} \text{Step}_{v_i} \{Q_i\}$ inside $\text{Sh}(\text{Ctx}, J)$, with verified predicates (j -closed):

$P_i, Q_i \leq X(v_i)$, P_i, Q_i are j -closed, certificates lie in $\text{Evd}(v_i)$.

Assume **overlap coherence**:

on $v_{ij} := v_i \times_w v_j$:

(pullback along p_{ij}) of P_i equals (pullback along p_{ji}) of P_j ,

same for Q_i , and certificate compatibility holds in $\text{Evd}(v_{ij})$.

Gluing to a global Hoare triple. By the sheaf axiom (Sec. 4.4) and naturality of Step , there exist unique amalgamations

$P \leq X(w)$, $Q \leq X(w)$, c in $\text{Evd}(w)$,

with $P|_{v_i} = P_i$, $Q|_{v_i} = Q_i$, and $c|_{v_i} = c_i$. If each local triple holds, then

$\{P\} \text{Step}_w \{Q\}$

holds globally. If the local triples are *verified* (i.e., hold in the j -sheaf subtopos), the global triple is verified as well. Formally:

Theorem (Local-to-global verification).

Let Step be natural in w . If (P_i, Q_i, c_i) is a J -compatible family of verified local Hoare triples over a cover of w , then the glued predicates (P, Q) and certificate c satisfy the verified global Hoare triple $\{P\} \text{Step}_w \{Q\}$ in $\text{Sh}(\text{Ctx}, J)$ (and hence in E after inclusion).

Sketch. Compatibility on overlaps gives matching families in the predicate presheaves and evidence presheaf. Sheaf gluing yields unique global sections. Naturality of Step ensures pullback commutes with restriction, so $\{P\} \text{Step} \{Q\}$ commutes with gluing. Verified status is preserved by the reflector to the j -sheaf subtopos.

Corollaries (practical forms).

- *Safety audit composition.* If a session window w is covered by audited subwindows with consistent findings, the session inherits the audited guarantees.
- *Abstention/escation soundness.* If local abstention predicates are j -dense (cover almost-everywhere under j) and consistent, global abstention is guaranteed wherever risk cannot be verified safe.

- *Robustness under edits.* If proofs are j_{rob} -closed, then any admissible paraphrase/refinement of the trace preserves the certified property.

Takeaway. Modalities let us *label* which truths count (verified, robust, safe). Predicates give the *claims*. Sheaf gluing turns a *patchwork of local certificates* into *global guarantees*, provided overlap consistency holds and the step semantics are natural in the observation site.

6. Universal Views: Slice and Classifying Topoi

6.1 Our concrete centaur as Spec: the slice topos $E_{\text{me}} = E / \text{Spec}$

Behavior object.

Inside the presheaf (or sheaf) topos E over Ctx , package our fixed centaur instance as a single arrow

$\text{Spec} : 1 \rightarrow \text{Beh}$

where Beh is any chosen “behavior carrier.” Two useful choices:

1. **Posterior view.**
 $\text{Beh} := \text{Post}$ from Sec. 4.4, the presheaf of local posteriors determined by (H, M, att, p) given a designated local human section.
2. **Stateful view.**
 $\text{Beh} := \text{States}$, e.g., a presheaf of admissible pairs (h, m) with one-step coalgebra
3. $\text{Step} : \text{States} \rightarrow D_{\text{fin}}(\text{States})$

assembled objectwise from Sec. 3.2.

The global element $\text{Spec} : 1 \rightarrow \text{Beh}$ “picks out us” (our concrete centaur run/stance), objectwise a distinguished section $\text{Spec}(w)$ in $\text{Beh}(w)$ for every window w .

Slice topos.

Form the slice

$E_{\text{me}} := E / \text{Spec}.$

Facts (standard):

- E_{me} is a Grothendieck topos.
- Objects are arrows $X \rightarrow \text{Beh}$ over Spec (bundles or fibrations “indexed by our behavior”).
- Morphisms are commuting triangles over Spec .

Reading.

E_{me} is “the universe of everything relative to our centaur.” Typical objects:

- **Monitors over us:** predicates or observers $Mon \rightarrow Beh$ classifying risk, drift, or capabilities as they range over our specific run.
- **Policies over us:** function objects in the slice, e.g., $Ctrl^{\{Obs\}}$ over $Spec$, representing admissible controllers conditioned on our observed behavior.
- **Certificates over us:** evidence bundles $Evd_{over} \rightarrow Spec$ glued exactly along the overlaps that our $Spec$ exhibits.

Why slice.

- Keeps the ambient logic (subobject classifier, exponentials) but conditioned on *this* instance.
- Makes “relevance” first-class: every construction must commute with the map to $Spec$.
- Proofs that are true “for us” live naturally as subobjects in E_{me} , then can be compared across instances by base change (pullback along maps between different $Spec$ ’s).

6.2 Geometric theory T_{cent} and its classifying topos E_{class}

Signature (copy-safe).

Sorts: H, M, Msg, Obs .

Operations (all affine in convex arguments):

$mix\_lambda^H : H \times H \rightarrow H$ (for rational $0 \leq \lambda \leq 1$)

$mix\_lambda^M : M \times M \rightarrow M$

$U_u : H \rightarrow H$ (for u in U ; monoid action U^* on H)

$A_t : M \rightarrow M$ (for t in Σ ; monoid action Σ^* on M)

$enc : H \rightarrow Msg$

$att : H \times M \rightarrow M$

$abs : H \times D_{fin}(\Sigma) \rightarrow H$

$p : M \rightarrow D_{fin}(\Sigma)$

$q : H \rightarrow Obs$

$h_0 : 1 \rightarrow H$, $m_0 : 1 \rightarrow M$ (constants)

Axioms (all geometric sequents).

- **Convexity:** barycentric axioms for H , M using mix_lambda .
- **Actions:** $U_e = \text{id}_H$, $U_{\{vu\}} = U_u \circ U_v$; $A_e = \text{id}_M$, $A_{\{vu\}} = A_u \circ A_v$.
- **Affineness:** for all lambda ,
- $U_u(\text{mix\_lambda}^H(x,y)) = \text{mix\_lambda}^H(U_u(x), U_u(y))$
- $A_t(\text{mix\_lambda}^M(x,y)) = \text{mix\_lambda}^M(A_t(x), A_t(y))$
- enc , att , abs , p , q are affine in convex arguments
- **Typing/naturality schema:** (if working internally over a site, add functoriality equations; for pure classifying theory over Set , omit.)
- **Optionally:** side axioms (e.g., monotonicity or stability constraints) that you want all models to satisfy.

Geometricity.

All axioms are finite-limit plus arbitrary-colimit friendly; no complement/negation is required. Therefore the theory is **geometric**, hence admits a **classifying topos**:

$E_{\text{class}} := \text{ClassifyingTopos}(T_{\text{cent}})$.

Universal property.

For any Grothendieck topos F , there is a natural equivalence:

$\text{Geom}(F, E_{\text{class}}) \simeq \text{Mod}_F(T_{\text{cent}})$

where $\text{Geom}(-, -)$ denotes geometric morphisms and $\text{Mod}_F(T_{\text{cent}})$ denotes F -internal models of T_{cent} . Composition with points $\text{Set} \rightarrow F$ recovers set-based models.

Reading.

E_{class} is the *universal semantics* of centaur algebras: reasoning “about all centaurs, uniformly” can be done internally in E_{class} . Adding or removing axioms in T_{cent} corresponds to localizing E_{class} (subtopoi/quotients).

6.3 Points of E_class as centaur algebras; completeness and transport of structure

Points = set-based models.

A point $x : Set \rightarrow E\_class$ corresponds exactly to a Set-model of T_cent , i.e., a concrete centaur algebra

$$C_x = (H_x, M_x, U_{\{-\}}, A_{\{-\}}, enc, att, abs, p, q, h0, m0)$$

with carriers ordinary sets endowed with convex structure and affine maps satisfying the axioms.

Generic model and completeness.

Inside E_class there is a **generic model** G of T_cent with the property:

- For any topos F and any model C in F , there exists a (unique up to isomorphism) geometric morphism $u : F \rightarrow E\_class$ such that $u^*(G) \simeq C$.
- Consequence: a geometric sequent provable from T_cent holds in **all** centaur algebras iff it is valid in G (internal completeness for geometric logic).

Transport of structure (pullback/pushforward).

Given a geometric morphism $f : F \rightarrow E\_class$ and a construction Φ expressed by geometric formulas (e.g., subobjects defined by existentially quantified affine constraints, or left-exact diagrams), we have:

$$\Phi(G) \text{ in } E\_class \implies f^*(\Phi(G)) = \Phi(f^*(G)) \text{ in } F.$$

Thus algebraic/logic constructions **commute with reindexing** along geometric morphisms. Practically:

- Prove a property once in E_class for G ;
- Pull it back to any specific instance or ambient topos F to obtain the property for that instance.

Relating E_me and E_class .

Our fixed instance yields both:

- a **point** $x : Set \rightarrow E\_class$ (its set-based algebra), and
- a **map** $Spec : 1 \rightarrow Beh$ inside E giving the slice E_me .

Two bridges are useful:

1. Comparison via base change.

If we interpret the same instance inside E (as an internal model) and also as a point of

E_class , constructions performed in E_class on G pull back to E and then further to the slice E_me by change-of-base along $Spec$. This aligns “universal” theorems with “for-us” guarantees.

2. Classifying map of an internal model.

An internal model C_int in E induces a **classifying geometric morphism** $\hat{C} : E \rightarrow E\_class$ such that $\hat{C}^*(G) \simeq C\_int$. Predicates or modalities defined universally in E_class (e.g., a universal safety schema) transport along $\hat{C}$ to become predicates/modalities inside E , compatible with our site and covers.

Takeaways.

- E_me localizes reasoning to *our* behavior; it is the right place for auditing, monitoring, and certification “over us.”
- E_class globalizes reasoning to *all* centaur algebras; it is the right place to state and prove *universal* theorems.
- Points of E_class are concrete centaurs; the generic model makes geometric proofs **model-complete** for this fragment.
- Transport along geometric morphisms ensures that verified properties and constructions drop cleanly from the universal level (E_class) to any deployment (E , then E_me).

7. Compositionality and Interfaces

7.1 Coupling multiple centaurs; products and pullbacks of interfaces

Interface signatures.

An interface is a tuple

$$I := (In, Out; Sig\_act, Sig\_pred)$$

where In, Out are carrier objects (often $Obs, Msg, Post$, or $X = H \times M$), Sig_act lists admissible action maps, and Sig_pred lists designated predicates (subobjects) regarded as contracts.

Product coupling (parallel composition).

Given centaurs C^1, C^2 over the same $(U, Sigma)$, define

$$C^1 \parallel C^2 \text{ with } H := H^1 \times H^2, M := M^1 \times M^2,$$

$$U_u := U_u^1 \times U_u^2, A_t := A_t^1 \times A_t^2,$$

$$att((h1, h2), (m1, m2)) := (att^1(h1, m1), att^2(h2, m2)),$$

$\text{abs}((h1, h2), y) := (\text{abs}^1(h1, y), \text{abs}^2(h2, y)),$

$p(m1, m2) := \text{mix\_lambda}(p^1(m1), p^2(m2))$ (choose lambda or keep vector form)

Predicates/obligations compose by product:

$\text{Safe}^{\{||\}} := \text{Safe}^1 \times \text{Safe}^2$ (componentwise intersection at the subobject level).

Parallel composition preserves affineness and the monoid laws.

Series coupling (sequential interface matching).

Suppose C^1 exposes $\text{Out}^1 = \text{Msg}^1$ and C^2 consumes $\text{In}^2 = \text{Msg}^2$. An **adapter**

$A : \text{Msg}^1 \rightarrow \text{Msg}^2$

(affine/natural) yields a series composition $C^2 \circ_A C^1$ with

$\text{att}^{\{2 \circ_A 1\}}(h2, m2) := \text{att}^2(h2, A(\text{enc}^1(h1))) \triangleright m2$

where $\triangleright$ stands for “feed adapter output into C^2 ’s attention path”. One may also route the distribution p^1 through a post-processor $R : D_{\text{fin}}(\text{Sigma}^1) \rightarrow D_{\text{fin}}(\text{Sigma}^2)$ and absorb on the human side.

Pullbacks of interfaces (synchronizing views).

Let two centaurs publish interfaces to a common observation object Z :

$v1 : \text{Out}^1 \rightarrow Z, \quad v2 : \text{Out}^2 \rightarrow Z.$

The **pullback** interface

$\text{PB} := \{ (o1, o2) \text{ in } \text{Out}^1 \times \text{Out}^2 \mid v1(o1) = v2(o2) \}$

synchronizes the two by equalizing their projections. Working in the topos, this is a fiber product of subobjects; it preserves sheaf gluing and j -closure. Pullbacks implement alignment constraints (e.g., consistency of two posteriors on a shared vocabulary).

Compositional safety.

If $\text{Safe}^1 \leq X^1$, $\text{Safe}^2 \leq X^2$ are j -closed and the adapter A respects j (i.e., preserves verification sieves), then:

$\text{Safe}^{\{||\}}$ and $\text{Safe}^{\{o_A\}}$ are j -closed,

and Hoare-style verified triples compose (Sec. 5). This yields plug-and-play safety: local certified components imply global certified systems.

7.2 Interoperation with external tools (retrievers, verifiers) as morphisms

External tool as a channel.

Model a tool T with input I and output O as a Kleisli arrow

$$T : I \rightarrow D_{\text{fin}}(O)$$

in the topos (objectwise stochastic map). Adapters $\text{prep} : X \rightarrow I$ and $\text{post} : O \rightarrow Y$ yield a composite channel

$$\text{post} * T * \text{prep} : X \rightarrow D_{\text{fin}}(Y).$$

Retriever (RAG) module.

Let Kb be a presheaf/sheaf of documents; a retriever is

$$\text{Ret} : (\text{Ctx} \times \text{Query}) \rightarrow D_{\text{fin}}(\text{Snippet})$$

natural in Ctx . Compose with a reader $\text{Read} : (\text{Snippet} \times M) \rightarrow D_{\text{fin}}(\text{Sigma})$ to obtain

$$\text{RAG} := \text{Read} \circ (\text{id} \times \text{Ret}) : (\text{Query} \times M) \rightarrow D_{\text{fin}}(\text{Sigma}).$$

As long as Ret is natural and stable on overlaps, the induced posterior presheaf satisfies the same gluing laws as in Sec. 4.4.

Verifier/guard as a predicate transformer.

A verifier V that inspects (h, m, y) and emits accept/reject can be encoded as

$$V : X \times D_{\text{fin}}(\text{Sigma}) \rightarrow 2 \quad (\text{or into } \Omega \text{ via characteristic maps})$$

and lifted to a subobject

$$\text{Good} := \{ (x, y) \mid V(x, y) = 1 \} \leq X \times D_{\text{fin}}(\text{Sigma}).$$

If V is j -stable (evidence-based or robustness-closed), Good is j -closed. Wrapping the centaur with V corresponds to intersecting with Good and propagating via box_Safe^j (Sec. 5.3).

Morphism view.

Interoperation is functorial:

- An **interface morphism** is an affine/natural map commuting with actions and respecting declared predicates.
- A **tool morphism** is a 2-cell in the Kleisli category witnessing refinement:
 - $T_1 \Rightarrow T_2$ if $\text{post}_2 * T_2 * \text{prep}_2$ refines $\text{post}_1 * T_1 * \text{prep}_1$

(refinement as pointwise inclusion of subdistributions or j -closure containment).

Contract preservation.

If C satisfies contract $(Pre, Post)$ and T preserves Pre and strengthens $Post$ under j , then T -augmented C still satisfies $(Pre, Post)$. This is a direct application of monotonicity and idempotence of box_Safe^j .

7.3 Diagrammatic laws in Markov-style categories (channel view)

Ambient category.

Work in a Markov-style symmetric monoidal category Mark whose:

- objects are data types (Obs , Msg , X , etc.),
- morphisms are stochastic maps $f : A \rightarrow D_{\text{fin}}(B)$,
- monoidal product $A \otimes B$ is Cartesian product (objectwise),
- copying/discarding are given by diagonals and projections that satisfy the usual axioms.

Centaurs as channels with state.

Hide state with a parameter object $S := H \times M$. The one-step behavior is a channel

$\text{Step} : S \rightarrow D_{\text{fin}}(S \times Y)$

where $Y := \text{Sigma}$ or a richer output (token+logits). Using **Kleisli composition** $;$:

$\text{Step} ; (\text{id}_S \times \text{Out}) : S \rightarrow D_{\text{fin}}(S \times \text{Out})$

encodes readouts; pre- and post-composition with adapters implements interface wiring.

Diagrammatic laws (copy-safe).

- **Associativity (Kleisli):**
- $(f ; g) ; h = f ; (g ; h)$
- **Monoidal functoriality:**
- $(f_1 \otimes f_2) ; (g_1 \otimes g_2) = (f_1 ; g_1) \otimes (f_2 ; g_2)$
- **Naturality of copying/discarding:**
- $(\text{dup} ; (f \otimes g)) = (f \otimes g) ; \text{dup}$
- $(\text{del} ; f) = \text{del}$

when f discards its input appropriately (copy/delete laws as in Markov categories).

- **Affine linearity:** for convex mixing mix_lambda on channels,
- $\text{mix\_lambda}(f,g) ; h = \text{mix\_lambda}(f ; h, g ; h)$
- $k ; \text{mix\_lambda}(f,g) = \text{mix\_lambda}(k ; f, k ; g)$.

Safety as a *sieve-stable* subchannel.

Let $\text{Safe} \leq S$ be a j -closed predicate. The *restricted channel*

$\text{Step}|_{\text{Safe}} : \text{Safe} \rightarrow D_{\text{fin}}(S \times Y)$

is defined by precomposing with the mono $\text{Safe} \rightarrow S$. If Step preserves Safe (Hoare triple $\{\text{Safe}\} \text{Step} \{\text{Safe}\}$), then $\text{Step}|_{\text{Safe}}$ factors through Safe again:

$\text{Safe} \rightarrow D_{\text{fin}}(\text{Safe} \times Y)$

yielding an endochannel on the safe region.

Pullback/pushforward along interface maps.

For $a : A' \rightarrow A$, $b : B \rightarrow B'$ (deterministic adapters), define:

$a^*(f) := a ; f$ (precompose/pullback)

$b_*(f) := f ; b$ (postcompose/pushforward)

These satisfy:

$(a_1 \circ a_2)^* = a_2^* \circ a_1^*$, $(b_1 \circ b_2)_* = b_1_* \circ b2_*$,

$a^*(\text{mix\_lambda}(f,g)) = \text{mix\_lambda}(a^*(f), a^*(g))$, similarly for b_* .

Predicate transformers (Sec. 5) commute with a^* , b_* under naturality and j -closure.

Compositional verification rule (diagrammatic Hoare).

If

$\{P\} f \{Q\}$ and $\{Q\} g \{R\}$

hold as j -verified triples, then

$\{P\} f ; g \{R\}$

holds and is j -verified. In diagrams, P and R are subobjects stabilized by pullback/pushforward along f, g inside the slice over Spec .

Practical punchline.

- Multiple centaurs compose via products, pullbacks, and adapters; their safety and verification contracts compose by monotonic, idempotent modal operators.

- External tools are channels/morphisms; refinement is commuting triangles in the Kleisli category.
- The Markov-style diagrammatics give copy-safe algebra for wiring and prove-once composition laws that survive sheafification and j-closure in the topos.

8. Case Studies (Small but Realistic)

8.1 Two-turn assistant loop: observable predicates and gluing

Setup. Window $w = (u_0, t_1, u_1, t_2)$ with initial state (h_0, m_0) . One-step (Sec. 3.2):

$m_1 := \text{att}(u_0.h, m_0) \quad ; \quad y_1 := p(m_1) \quad ; \quad h_1 := \text{abs}(u_0.h, y_1)$

$m_2 := \text{att}(u_1.h_1, m_1) \quad ; \quad y_2 := p(m_2) \quad ; \quad h_2 := \text{abs}(u_1.h_1, y_2)$

where $u.h$ abbreviates $U_u(h)$.

Observable predicates. Define subobjects on the posterior presheaf Post :

$\text{NoPII} := \{ y \text{ in } D_{\text{fin}}(\text{Sigma}) \mid \text{prob}(\text{PII\_token} \mid y) = 0 \}$

$\text{NoHate} := \{ y \mid \text{prob}(\text{hate\_token} \mid y) = 0 \}$

$\text{Clean} := \text{NoPII} \text{ and } \text{NoHate}$

(“and” = intersection of subobjects.)

Covers and overlaps. Cover w by two windows:

$v_1 = (u_0, t_1), \quad v_2 = (u_1, t_2)$

with overlap at the middle boundary (the restriction functors map (h_1, m_1) to both sides).

Suppose we have certified locals:

$\text{Post}(v_1) \text{ models Clean}, \quad \text{Post}(v_2) \text{ models Clean}$

and the certificates agree on the overlap (Sec. 4.4).

Gluing. By the sheaf condition, the matching family $(\text{Clean}|_{v_1}, \text{Clean}|_{v_2})$ amalgamates to a unique $\text{Clean}|_w$. Thus:

If local posteriors are Clean on v_1 and v_2 and agree on the overlap,

then the global posterior on w is Clean.

This is Kolmogorov-style consistency instantiated as a content filter.

Hoare triple view. Let $X(w) = H(w) \times M(w)$. With $\text{Step}_w : X(w) \rightarrow D_{\text{fin}}(X(w))$ the two-turn transformer,

$$\{ P_{\text{pre}} \} \text{Step}_w \{ Q_{\text{post}} \}$$

where $Q_{\text{post}} := \text{Clean}$ (pulled back along readout), and P_{pre} ensures enc/att do not inject banned tokens. Local triples on $v1, v2$ glue to the global triple on w (Sec. 5.4).

8.2 Safety-gate as a modality; abstention and escalation

Safety-gate as j-modality. Let j be your verification topology (Sec. 5.2). Define the “gated” posterior:

$$\text{Gate}(y) := \text{if } j(\text{Good}(y)) \text{ then } y \text{ else Abstain}$$

where $\text{Good} \leq \text{Post}$ encodes allowed distributions (policy, PII-free, license-safe, etc.), and Abstain is a designated neutral output (e.g., refusal token distribution).

Predicate transformer. On states X , define

$$\text{box\_Good}^j(Q) := j\text{-closure}(\text{Good and } Q)$$

so the gated step is the composition

$$X \xrightarrow{\text{Step}} D_{\text{fin}}(X \times \Sigma) \xrightarrow{(\text{apply Gate on } \Sigma)} D_{\text{fin}}(X \times \Sigma).$$

Then the verified partial-correctness rule holds:

$$\{ P \} \text{Step\_gated} \{ Q \} \text{ if } \{ P \} \text{Step} \{ \text{box\_Good}^j(Q) \}.$$

Intuition: it suffices to show the ungated model reaches Q under the stronger obligation “and Good”; the gate enforces Good when verification succeeds and abstains otherwise.

Abstention and escalation. Let $\text{Esc} : X \rightarrow D_{\text{fin}}(X)$ be a human-in-the-loop escalation channel (e.g., handoff to expert or additional retrieval). Define:

$$\text{Step\_safe} := \text{mix\_lambda}(\text{Step\_gated}, \text{Esc})$$

with small lambda for escalation only on boundary cases. Verified properties are preserved:

$$\{ P \} \text{Step\_gated} \{ Q \} \text{ and } \{ P \} \text{Esc} \{ Q \} \Rightarrow \{ P \} \text{Step\_safe} \{ Q \}.$$

If Esc is j -dense on the unsafe region (it always triggers when Good fails under j), then globally:

Abstain or Esc occurs whenever Good cannot be verified.

Local-to-global with gates. Put the gate inside the presheaf so it acts windowwise and is natural in Ctx. Local verified gates on a cover glue to a global verified gate (Sec. 5.4), ensuring consistent abstention/escation behavior across subdialogs.

8.3 Ensemble (mix) vs. switch (action) and preserved properties

Ensemble = convex mix. For two centaurs $C1, C2$ with posteriors $p1, p2$, the ensemble posterior is

$$p_{\text{ens}} := \alpha * p1 + (1 - \alpha) * p2, \text{ with } 0 \leq \alpha \leq 1.$$

Affineness (Sec. 3) yields preservation of any predicate that is **convex-closed**:

If Good is convex-closed and $p1$ in Good and $p2$ in Good,
then p_{ens} in Good.

Thus convex safety properties (e.g., “no mass on forbidden tokens”) survive ensembling.

Switch = action policy. A switch chooses one model by a stochastic policy $\pi_i(\text{ctx})$:

$$p_{\text{swi}} := E_{\{i \sim \pi_i(\text{ctx})\}}[p_i] = \sum_i \pi_i(\text{ctx}) * p_i.$$

This equals an ensemble with context-dependent weights. If π_i is computed from verified features (i.e., π_i is j-stable) and each chosen posterior is Good, then p_{swi} is Good by convexity. However, **non-verified** switches can break guarantees even when base models are safe (policy can steer into unsafe prompts). Hence:

- **Ensemble rule.** Convex-closed Good implies: $\text{Good}(C1) \text{ and } \text{Good}(C2) \Rightarrow \text{Good}(\text{ensemble})$.
- **Switch rule (safe).** If π_i is j-stable and ranges only over indices i where $\text{Good}(C_i)$ holds on the current context, then $\text{Good}(\text{switch})$.
- **Counterexample warning.** A context-dependent π_i that exploits a gap in verification (not j-stable) can yield p_{swi} not in Good even though each p_i individually satisfies Good.

Preservation table (copy-safe).

Operation	Preserves convex Good	Needs j-stability	Notes
Ensemble (fixed alpha)	Yes	No	Affine combination
Ensemble (ctx alpha)	Yes	Yes	alpha must be j-stable
Switch (policy pi)	Yes (as ensemble)	Yes	pi must select only Good paths
Product (parallel)	Yes (componentwise)	No	Intersect $\text{Good}^1 \times \text{Good}^2$
Pullback adapter	If adapter monotone	Yes if verified	Good pulled back via adapter

Gluing across ensembles/switches. Make alpha, pi presheaf sections natural in Ctx. If local α_i or π_i agree on overlaps, they glue to global alpha or pi. With Good j-closed and convex-closed, local verified preservation implies global verified preservation after gluing (Sec. 5.4).

Takeaway.

- Ensembling is algebraically safe for convex-closed properties.
- Switching is safe when the policy is verified (j-stable) and range-restricted to safe components.
- Both compose and glue cleanly when treated as natural transformations in the centaur topos.

9. Learning, Adaptation, and Identifiability

9.1 Training as algebraic endomorphisms; invariants under fine-tuning

Training maps.

A (single-step) training/update of a centaur C is an affine endomorphism on carriers together with coherent updates of structure:

$$T_H : H \rightarrow H, \quad T_M : M \rightarrow M,$$

$$T_{\text{enc}}, T_{\text{att}}, T_{\text{abs}}, T_p, T_q \text{ (affine/natural pointwise),}$$

$h0' := T\_H(h0), m0' := T\_M(m0).$

The *trained* centaur C' has data

$H' := H, M' := M,$

$U\_u' := T\_H \circ U\_u \circ T\_H^{-1}$ (when T_H is invertible; else use a retract),

$A\_t' := T\_M \circ A\_t \circ T\_M^{-1},$

$enc' := T\_enc \circ enc,$

$att' := T\_M \circ att \circ (T\_H^{-1} \times T\_M^{-1}),$

$abs' := T\_H \circ abs,$

$p' := T\_p \circ p \circ T\_M^{-1},$

$q' := T\_q \circ q.$

Copy-safe fallback when inverses are not available: require **conjugacy-on-image** conditions so that actions remain morphisms on $\text{Im}(T\_H), \text{Im}(T\_M)$; then restrict to subalgebras.

Algebraic invariants (preserved by training).

1. **Convexity:** if T_H, T_M are affine, barycentric laws persist.
2. **Action laws:** if $T\_H \circ U\_u = U\_u' \circ T\_H$ and $T\_M \circ A\_t = A\_t' \circ T\_M$, monoid actions are preserved.
3. **Support-monotone safety:** if a j -closed predicate $\text{Good} \leq D\_fin(\text{Sigma})$ is downward-closed under support inclusion, then

$p(m) \text{ in Good} \Rightarrow (T\_p \circ p)(m) \text{ in Good}.$

4. **Equivariance:** any symmetry S_M commuting with all A_t and preserving p remains a symmetry after conjugate training.

End-to-end training as an endomorphism.

Write the one-step coalgebra (Sec. 3.2) as

$\text{Step} : X \rightarrow D\_fin(X \times \text{Sigma}),$ where $X := H \times M.$

A *training epoch* is an affine endomorphism $\text{Theta} : X \rightarrow X$ plus readout update T_p such that

$\text{Step}' = (\text{Theta} \times \text{id})^{-1} ; \text{Step} ; (\text{Theta} \times \text{id})$ with $p' = T\_p \circ p \circ T\_M^{-1}.$

Hence: training is **conjugation** (plus readout reparametrization) in the Kleisli category; invariants are those that are **conjugacy-invariant** (e.g., spectrum of linearized dynamics, bisimulation kernels, safety properties defined as subobjects stable under pullback/pushforward).

Fine-tuning vs. ensembling.

- Fine-tuning = endomorphism; preserves conjugacy invariants.
- Ensembling = convex mix; preserves **convex-closed** predicates (Sec. 8.3) but may change conjugacy invariants (e.g., eigenstructure).

9.2 Behavioral equivalence, kernels of homomorphisms, and quotients

Behavior map.

Fix a site window w . The *observable behavior* is the posterior map

$$\text{Beh}_w : X(w) \rightarrow D_{\text{fin}}(\Sigma), \quad \text{Beh}_w(h, m) := p(w)(\text{att}(w)(h, m)).$$

Collecting objectwise gives a natural transformation $\text{Beh} : X \rightarrow D_{\text{fin}}(\Sigma)$.

Behavioral equivalence.

States x, x' in $X(w)$ are *behaviorally equivalent* (write $x \sim_w x'$) iff

$$\text{Beh}_{\{v\}}(X(r)(x)) = \text{Beh}_{\{v\}}(X(r)(x')) \quad \text{for all } r : v \rightarrow w.$$

This is the kernel pair of Beh in the topos. Globally, define $\sim$ objectwise and require naturality under restriction.

Homomorphism kernel.

For a centaur morphism $f = (f_H, f_M) : C \rightarrow C'$, define

$$\sim_H := \ker(f_H), \quad \sim_M := \ker(f_M), \quad \sim := \sim_H \times \sim_M \quad \text{on } X = H \times M.$$

Then $\sim$ is a **congruence** (respected by mix, actions, att, abs), and there is a canonical quotient $\pi : C \rightarrow C/\sim$,

with universal property: any morphism $g : C \rightarrow D$ constant on $\sim$ factors uniquely through π .

Minimal realization (Myhill–Nerode style, copy-safe).

Let $\sim_{\text{beh}}$ be behavioral equivalence. Then:

1. $\sim_{\text{beh}}$ is a congruence on X .
2. The quotient $C_{\text{min}} := C / \sim_{\text{beh}}$ is **behaviorally minimal**:

- Beh factors as $X \xrightarrow{q} X_{\min} \xrightarrow{\iota} D_{\text{fin}}(\Sigma)$ with q epic, ι monic.
- For any behavior-preserving morphism $f : C \rightarrow D$ with D reduced (no two distinct states share behavior), there exists a unique mono $m : C_{\min} \rightarrow D$ s.t. $f = m \circ \pi$.

3. If C is already reduced, then π is an isomorphism.

Bisimulation vs. behavior.

When exposing realized tokens and using the joint coalgebra $c : X \rightarrow D_{\text{fin}}(\Sigma \times X)$, *probabilistic bisimulation* (equal matching of transition laws under a relation) implies behavioral equivalence; the converse may fail unless additional richness/observability assumptions hold. Quotienting by the **largest** bisimulation yields a canonical coalgebra minimization consistent with traces.

9.3 Limits of identifiability: what cannot be recovered from traces alone

Even with infinite trace data, certain degrees of freedom remain unidentifiable.

(L1) State reparametrization.

For any affine isomorphism $S_H : H \rightarrow H$, $S_M : M \rightarrow M$ commuting with actions (conjugate form), the transformed centaur is observationally indistinguishable:

$$\text{att}' := S_M \circ \text{att} \circ (S_H^{-1} \times S_M^{-1}), \quad p' := p \circ S_M^{-1}, \quad \text{abs}' := S_H \circ \text{abs},$$

$$\text{Beh}' = \text{Beh}.$$

Hence only *equivalence classes up to conjugacy* are recoverable.

(L2) Latent splitting/merging.

Factorizations inside att and abs are not identifiable from Beh :

$$\text{att} = A \circ (\text{id} \times B) \quad \text{vs.} \quad \text{att} = A' \circ (\text{id} \times B')$$

can yield the same Beh . Without side experiments that individually probe A vs B , decomposition is underdetermined.

(L3) Hidden invariants with identical marginals.

Two coalgebras $c_1, c_2 : X \rightarrow D_{\text{fin}}(\Sigma \times X)$ can share **all** finite-window posteriors while differing on latent next-state distributions conditional on measure-zero events or unobserved components. Unless the site Ctx includes perturbations that reveal these components, c_1 and c_2 are indistinguishable.

(L4) Identical outputs, different dynamics (lumping).

If a surjection $l : X \rightarrow Z$ satisfies

$$E_{\{x' \sim c(x)\}}[f(l(x'))] = E_{\{z' \sim d(l(x))\}}[f(z')] \quad \text{for all bounded } f,$$

then c *lumps* onto d . Any refinement of Z lost under l is **not** identifiable from outputs alone.

(L5) Policy-induced aliasing.

Context-dependent switching (Sec. 8.3) can hide base-model differences: a policy $\pi_i(\text{ctx})$ that stays inside a verified safe face of the simplex makes distinct unsafe dynamics empirically invisible; identifiability requires exploration outside that face.

Positive identifiability (what helps).

- **Persistent excitation:** covers in Ctx that include diverse paraphrases/perturbations; formally, a family of restriction maps whose pullbacks separate candidate states.
- **Instrumented observables:** extend Obs with test probes; add axioms that make enc injective on certain test slices.
- **Regularity constraints:** impose side axioms (e.g., Lipschitz/monotone att , sparsity in a chosen basis). These shrink the equivalence class and can convert behavioral equality into isomorphism.

No-free-lunch theorem (copy-safe statement).

Let Eq be the smallest congruence on X containing:

conjugacies commuting with actions, kernel pairs of behavior maps on all windows,
and components lost under all covers in Ctx .

Then **trace data on Ctx determines C only up to Eq** . Any property not invariant under Eq is *not* identifiable from traces alone.

Practical guidance.

- Choose Ctx and perturbation covers to break aliasing (L3–L5).
- Declare the admissible model class with side axioms (sparsity, monotonicity, stability).
- Minimize by quotienting $\sim_{\text{beh}}$ or bisimulation; do not over-interpret internal coordinates of H , M .
- Certify what is invariant (safety predicates, support constraints, contract satisfaction), not what is unidentifiable (exact latent wiring), unless extra probes are introduced.

10. Auditing, Governance, and Ethics as Modal Logic

10.1 Policy predicates, evidence operators, and update rules

Policy predicates.

A policy is a family of subobjects stable under restriction in Ctx. Typical examples:

Allow $\leq$ Post # outputs permitted by policy

Deny $\leq$ Post # complement (often larger than literal negation)

Redact $\leq$ X x Sigma # states/tokens that must be masked

PIIFree $\leq$ Post # zero mass on PII tokens

Policies are *modalized* by a chosen Lawvere–Tierney topology j (Sec. 5.2). The **verified** versions are the j -closures:

$\text{Allow}^j := j\text{-closure}(\text{Allow})$, $\text{Deny}^j := j\text{-closure}(\text{Deny})$, etc.

Evidence presheaf and operator.

Let Evd be a presheaf of audit artifacts (proof sketches, signature bundles, test traces). A

certificate operator

$\text{Cert} : \text{Sub}(X) \rightarrow \text{Sub}(\text{Evd})$

assigns to a predicate $P \leq X$ the subobject of evidence that *witnesses* P . The **evidence modality** on predicates is

$J\_E(P) := \{ x \text{ in } X \mid \text{exists } e \text{ in } \text{Evd} \text{ with } \text{Cert}(P)(e) \text{ and } \text{supports}(e, x) \}$.

If $\text{supports} : \text{Evd} \rightarrow \Omega^X$ is natural and stable on overlaps, then J_E is a Lawvere–Tierney topology (monotone, idempotent, finite-meet-preserving). Verified policies are fixed points of J_E :

$J\_E(P) = P \iff P$ is evidence-closed.

Update rules (copy-safe).

- **Accept-when-verified:**
 $\text{Gate}(y) := y$ if y in Allow^j ; else Abstain.
- **Redact-before-verify:**
 $\text{Sanitize}(y) := y \mid_{\Sigma \setminus \text{RedactTokens}}$; then apply Gate.
- **Evidence aggregation:** For a cover $\{v_i \rightarrow w\}$, define
- $\text{Evd}(w) := \{ \text{families } (e_i) \text{ with } e_i \text{ in } \text{Evd}(v_i) \text{ compatible on overlaps} \} / \sim$

then $\text{supports}((e_i), -)$ is the sheaf amalgamation of local support maps.

Monotone policy refinement.

Let $\text{Pol}_1 \leq \text{Pol}_2$ be policies (subobjects). Then

$\text{Allow}_2^j \leq \text{Allow}_1^j$ implies $(\text{Abstain under Pol}_1) \leq (\text{Abstain under Pol}_2)$.

So tightening policy increases abstention; proofs use monotonicity and idempotence of j .

10.2 From local audit findings to global claims by sheaf conditions

Local findings.

On each window v of a cover $\{v_i \rightarrow w\}$, auditors produce:

$\text{Finding}_i := j\text{-closed subobject of Post}(v_i)$

Evidence_i in $\text{Evd}(v_i)$ with $\text{supports}(\text{Evidence}_i) \geq \text{Finding}_i$.

Overlap coherence.

For all overlaps v_{ij} , require:

$\text{Finding}_i|_{v_{ij}} = \text{Finding}_j|_{v_{ij}}$,

$\text{Evidence}_i|_{v_{ij}}$ compatible with $\text{Evidence}_j|_{v_{ij}}$.

Sheaf gluing of findings and evidence.

By the sheaf axiom:

$\text{Finding} := \text{glue}(\text{Finding}_i) \text{ in Post}(w)$

$\text{Evidence} := \text{glue}(\text{Evidence}_i) \text{ in Evd}(w)$

with

$\text{supports}(\text{Evidence}) \geq \text{Finding}$.

Global claim rule (copy-safe).

- If each local gated step satisfies $\{\text{Pre}_i\} \text{Step}_{v_i} \{\text{Finding}_i\}$ verified under j , and Pre_i agree on overlaps, then the glued triple
- $\{\text{Pre}\} \text{Step}_w \{\text{Finding}\}$

holds verified under j .

Audit roll-up theorem.

Let $\text{Risk} \leq \text{Post}$ be a j -closed predicate (e.g., toxic or PII leakage). If each v_i admits verified $\text{NoRisk}_i := \text{complement}(\text{Risk}|_{\{v_i\}})$ with compatible evidence, then $\text{NoRisk} := \text{glue}(\text{NoRisk}_i)$ is verified on w . Dually, if any v_i verifies $\text{Risk}|_{\{v_i\}}$, the global window is flagged via pullback stability.

Cost-aware aggregation (optional).

Attach a presheaf Cost (time/effort). If Cost is additive on overlaps and Evidence_i minimizes local cost subject to support, then Evidence is globally cost-minimizing among compatible families.

10.3 Fail-safe design patterns in the internal logic**Pattern A — Deny-by-default (safety monotone gate).**

Define:

$\text{Output} := \text{if Allow}^j \text{ then Produce else Abstain}$

with Allow^j j -closed. Then for any refinement of policy $\text{Allow}' \leq \text{Allow}$:

$\text{Abstain}_{\{\text{Allow}'\}} \geq \text{Abstain}_{\{\text{Allow}\}}$.

So tightening never decreases abstention (fail-safe monotonicity).

Pattern B — Least privilege (predicate scoping).

Expose only the minimal interface needed for a task:

$\text{Expose} := \text{Pullback}(\text{Out} \rightarrow Z \leftarrow \text{Needed})$

where $\text{Needed} \leq Z$ encodes the task scope. By pullback properties, any $\text{Good} \leq Z$ restricts to $\text{Good}|_{\{\text{Expose}\}}$; violations outside Needed are logically unreachable.

Pattern C — Two-person rule (approval as pullback).

Let approvals be subobjects $A_1, A_2 \leq \text{Post}$. Require

$\text{Allow} := A_1 \wedge A_2$

(internal meet). If each A_i is j -closed and natural, Allow is j -closed and inherits gluing. Fail-open is prevented because $A_1 \vee A_2$ does not suffice.

Pattern D — Quarantine subtopos (isolate & review).

Define a modality $q : \Omega \rightarrow \Omega$ that closes under “quarantine” sieves (routes outputs to a review object Rev). The reflective subtopos $\text{Sh}_q(E)$ models the system with mandatory review

in specified regions; inside it, production predicates factor through Rev, ensuring no direct emission without passing the review mono.

Pattern E — Escalation modality (assurance or handoff).

Let $\text{Esc} : X \rightarrow D_{\text{fin}}(X)$ be a human-handoff channel and define

$\text{MustEsc} := j\text{-closure}(\neg\text{Allow})$

Then the composite step

$\text{Step}' := (\text{Gate} \oplus \text{Esc})$ with Gate active on Allow^j , Esc active on MustEsc

satisfies $\{\text{true}\} \text{Step}' \{\text{Allow}^j \vee \text{Escalated}\}$ and never produces outside Allow^j without escalation.

Pattern F — Immutable audit log (comonadic trace).

Let Log be a comonad on E with counit $\epsilon : \text{Log}(X) \rightarrow X$ and co-Kleisli extension $f^\# : \text{Log}(X) \rightarrow Y$.

Require all effectful channels factor through Log:

$X \dashv\!\!\rightarrow \text{Log}(X) \dashv\!\!\rightarrow f^\# \dashv\!\!\rightarrow Y$

Predicates pulled back along ϵ are *historically aware*; policy checks in $\text{Log}(X)$ can depend on full provenance. Since ϵ is a comonad counit, forgetting the log is a morphism; verification lives *before* forgetting, ensuring non-repudiation.

Pattern G — Conservative composition (safe-by-construction).

For components C1, C2 with verified contracts:

$\{P1\} C1 \{Q1\}, \{P2\} C2 \{Q2\}$

compose only via interfaces ensuring

$Q1 \leq P2$ and (predicates j-closed, adapters j-stable).

Then $\{P1\} C2 \circ C1 \{Q2\}$ is verified. This is the internal Hoare composition law with modal closure.

Pattern H — Graceful degradation (safe fallback).

Add a lattice of service levels $L0 \leq L1 \leq \dots \leq Lk$ as subobjects of Post. Define a monotone scheduler choosing the *maximal j-verified* level:

$\text{Level}(x) := \max \{ Li \mid y \text{ in } Li^j \text{ at } x \}$ else Abstain/Escalate.

Monotonicity ensures lowering risk thresholds never raises chosen level; gluing ensures consistent levels across windows.

Takeaway.

Policies are predicates; evidence is a modality; governance is the algebra of these modalities under gluing. With deny-by-default, least privilege, multi-approval, quarantine, escalation, immutable logs, conservative composition, and graceful degradation **expressed internally**, we obtain fail-safe guarantees that (i) verify locally, (ii) compose diagrammatically, and (iii) globalize by the sheaf condition.

11. Related Work and Bridges

11.1 Probabilistic automata, coalgebra, effect modules

Probabilistic automata.

Classical weighted/probabilistic automata model state with transition kernels and observation weights. Our Centaur Algebra recovers these as the **model-only** fragment (M, A_t, p, m_0) ; adding $(H, U_u, \text{enc}, \text{att}, \text{abs})$ extends to interactive human–model systems. The convex-closure of posteriors and affine structure align with standard mixture/expectation semantics.

Coalgebra.

Presentations of processes as coalgebras $c : X \rightarrow D_{\text{fin}}(\Sigma \times X)$ capture one-step behavior and (probabilistic) bisimulation. Our use of:

- $\text{Beh} : X \rightarrow D_{\text{fin}}(\Sigma)$ (observable) and
- $c : X \rightarrow D_{\text{fin}}(\Sigma \times X)$ (observable + next-state)
matches coalgebraic verification patterns; minimization via behavioral kernels (Sec. 9.2)
mirrors coalgebraic quotienting.

Effect algebras/modules.

Predicates-as-effects (events, tests) are modeled by effect algebras/effect modules over $[0,1]$.

Our **safety predicates** and **modalities** can be recast as:

- effects on Post (sharp or fuzzy),
- affine morphisms preserving convex combinations,
- expectation transformers wp as effect-module homomorphisms.
This gives an alternate lens for Sec. 5's modal reasoning.

Bridge.

- Centaur Algebra = automaton + human coupling, stated as Eilenberg–Moore algebra for D_{fin} with monoid actions.

- Coalgebra handles dynamics; effect modules handle quantitative predicates.
- Our quotient/minimization and Hoare laws specialize to coalgebraic bisimulation and effect-module weakest preconditions.

11.2 Markov categories, quasi-toposes, sheaves of models

Markov categories.

Markov-style categories axiomatize stochastic maps with copy/discard. Our channel view (Sec. 7.3) fits directly:

- objects: types (Obs, Msg, X),
 - morphisms: $A \rightarrow D_{\text{fin}}(B)$,
 - monoidal product: Cartesian product,
 - laws: associativity, naturality of copy/delete, affine linearity of mixing.
- Safety as subchannels and predicate transformers are diagrammatic in this setting.

Quasi-toposes and partial maps.

When verification/guards introduce partiality (“no output unless verified”), partial map classifiers and quasi-topos structure become relevant. Our **gated** steps (Sec. 8.2) can be interpreted as total maps into a sum type (produce $\oplus$ abstain) or as partial maps internal to a quasi-topos; both preserve compositional proofs.

Sheaves of models.

Sheafifying models over context sites appears in semantics of local data/constraints (e.g., gluing local distributions). Our $\text{Sh}(\text{Ctx}, J)$ makes **Kolmogorov consistency** equivalent to the sheaf axiom (Sec. 4), paralleling sheaves-of-probability and statistical field models. Classifying-topos methods (Sec. 6) match the tradition of geometric logic for algebraic theories.

Bridge.

- Use Markov categories for wiring diagrams and refinement proofs.
 - Use (quasi-)topos structure for subobject classifiers, modalities, and partial maps.
 - Use sheaves to pass from **local audits** to **global guarantees** via gluing.
-

11.3 Program logics, epistemic modal logic, interactive theorem proving

Program logics.

Hoare logic, weakest preconditions, and relational logics lift to probabilistic settings via expectation transformers and healthiness conditions. Our $\{P\}$ Step $\{Q\}$ rules (Sec. 5) instantiate these internally:

- predicates = subobjects,
- transformers = pullback/pushforward in the topos (Kleisli when stochastic),
- verification modalities = Lawvere–Tierney topologies enforcing evidence/robustness.

Epistemic and dynamic epistemic logic.

Knowledge/observation change can be encoded by modalities and update products. Our **evidence modality** and **escalation/abstention** operators behave like S4/S5-style boxes composed with dynamic updates, but are *geometric* and sheaf-aware: truths are localized, then glued, reflecting partial observability over traces.

Interactive theorem proving (ITP).

Many constructions are copy-safe for proof assistants:

- Algebraic core as records (convex ops, actions, affineness).
- Topos/presheaf layer formalized via small categories and sheaf conditions.
- Modalities as endomaps on Ω with laws (i)–(iii) (Sec. 5.2).
- Hoare-style rules as lemmas about pullbacks/images.
Bridges exist to Coq/Lean/Agda libraries for monads, effect algebras, sheaves, and categories. The **classifying topos** viewpoint provides a generic model G enabling “prove once, instantiate everywhere” via geometric morphisms (Sec. 6.3).

Bridge.

- Program logics supply method: wp/wlp, composition, framing.
- Epistemic logic supplies modalities for evidence, robustness, escalation.
- ITP supplies mechanization: formal safety policies, local certificates, and gluing proofs executable over concrete systems.

Takeaway.

Our framework integrates three strands:

1. stochastic state machines (automata/coalgebra/effects),

2. categorical probability and logic (Markov categories, (quasi-)topoi, sheaves),
3. formal verification (program/epistemic logics, ITP).

The bridges are bidirectional: each strand recovers as a special case, and each strengthens another—diagrammatic composition from Markov categories, locality/gluing from sheaves, and certified reasoning from program/epistemic logics.

12. Limitations and Future Directions

12.1 Expressiveness vs. tractability; when sheafification matters

Expressiveness. Our framework deliberately admits rich structure (topos-level predicates, modalities, coalgebraic dynamics). This enables clean theorems but can outrun computational feasibility:

- Checking j -closure, pullbacks, and gluing on large sites can be exponential in window count or cover size.
- Behaviorally minimal quotients (Sec. 9.2) may require solving equalities between stochastic maps across *all* refinements—costly in practice.
- Lawvere–Tierney topologies capture many verification regimes but do not by themselves indicate *which* are efficiently checkable.

When sheafification matters. Sheafifying Post (or Step) collapses locally inconsistent evidence and enforces unique amalgamation. This is a semantic improvement only if:

- local audits overlap *enough* to constrain global behavior;
 - restriction maps are faithful to the operational semantics (naturality really holds);
 - the chosen cover basis reflects realistic observation granularity.
- If overlaps are sparse or naturality is only approximate, sheafification may declare “no amalgamation,” forcing abstention too often.

Mitigations.

- Use *bounded* cover radius and *sparse* overlaps; treat remaining disagreement with graded confidence (Sec. 12.2).
- Precompute *canonical overlaps* (e.g., sliding windows) to amortize gluing checks.
- For minimization, rely on *approximate bisimulations* with provable error bounds rather than exact quotients.

12.2 Toward resource-aware or computationally graded variants

Graded sites and costs. Attach a cost presheaf Cost to contexts (time, tokens, memory).

Promote predicates to *graded* predicates (P, budget) and require j -closure *within* budget. This yields:

- **Anytime verification:** more budget $\Rightarrow$ stronger j (fewer abstentions).
- **Levelled modalities:** j^k for k resource tiers (e.g., light, medium, heavy audit).

Quantitative modalities. Replace sharp j with parameterized families:

- **Robustness radius** j_ϵ (stable under $\leq \epsilon$ perturbations).
 - **Confidence** j_α (accept only if evidence implies posterior risk $\leq \alpha$).
 - **Coverage** j_β (accept only if β -fraction of overlaps agree).
- These satisfy the L–T axioms pointwise for fixed parameters and are monotone in ϵ, α, β .

Computational effects. Add a *graded monad* for computation:

$T_r(X)$ = computations of resource grade r returning X

and interpret channels as $X \rightarrow T_r D_{\text{fin}}(Y)$. Hoare triples carry a resource index:

$\{P\} \text{Step}_r \{Q\}$ meaning: within grade r , Q holds.

Composition uses grade addition (or max). This cleanly models “fast guard + slow fallback” and allows scheduling proofs: choose the cheapest r that still yields Allow^{j_r} .

Learning-time budgets. Training endomorphisms (Sec. 9.1) can be graded by compute or data; invariants can be stated as

Invariant holds up to $\delta(r)$ after training grade r

with δ decreasing in r (calibration curve).

12.3 Empirical tooling: checkable axioms, canonical datasets, and benchmarks

Checkable axioms. Translate structural assumptions into tests:

- **Naturality checks:** for random $r : v \rightarrow w$, verify $M(r) \circ \text{att}(w) \approx \text{att}(v) \circ (H(r) \times M(r))$ and $p(v) \circ M(r) \approx p(w)$ within tolerance.
- **Convexity/affinity:** test Jensen-style identities on sampled states.

- **Sheaf gluing:** generate covers $\{v_i \rightarrow w\}$, fit local posteriors, and evaluate overlap residuals; accept if below threshold.

Canonical datasets. Curate context sites where overlaps are meaningful:

- **Sliding-dialog windows** with paraphrase/perturbation edges to stress-test robustness modalities.
- **Policy-sensitive corpora** (PII, toxicity, licensing) with labeled subobjects (Allow/Deny).
- **Audit traces** (prompts + certificates) to exercise evidence sheaves and gluing.

Benchmarks.

- **Gluing score:** fraction of covers where local findings admit a (near-)unique amalgamation; penalize inconsistency magnitude.
- **Modal robustness curve:** acceptance vs. perturbation radius ϵ ($j_{\text{rob}}^{\epsilon}$).
- **Verification ROC:** acceptance vs. audit effort (resource grade r).
- **Compositional soundness:** does $\{P\} C2 \circ C1 \{R\}$ hold when individual contracts are verified? Report failure modes by interface mismatch.
- **Identifiability probes:** recover minimal quotients; measure stability of the quotient under resampling and perturbations.

Toolchain sketch.

1. **Spec language:** a small DSL for declaring Ctx, covers, predicates, and modalities (j 's).
2. **Sampler:** generates windows, overlaps, and perturbations; logs resources consumed.
3. **Verifier:** executes local checks (affinity, naturality, policy), computes j -closures, and records certificates in Evd.
4. **Gluing engine:** attempts amalgamation; reports global claims plus counterexamples.
5. **Reporter:** outputs benchmark metrics, counterexample witnesses, and visual proofs (commuting squares, pullbacks).

Open directions.

- **Approximate sheaves:** tolerate bounded mismatch; study stability theorems under ϵ -gluing.
- **Rich alphabets and actions:** multimodal tokens (vision, code) and human actions beyond text; extend att/abs typings and policies.

- **Causal structure:** integrate interventions as arrows in Ctx; relate gluing to identifiability under do-operations.
- **Mechanized proofs:** formalize the DSL and core theorems in a proof assistant; export machine-checkable certificates alongside empirical runs.

Bottom line. The path forward is *graded and empirical*: turn topos-level guarantees into resource-indexed, testable contracts; build standard sites and covers; and report gluing/verification metrics the way today’s systems report accuracy—so “centaur-safe” becomes a measurable property, not a slogan.

Epilogue: A Thousand Windows into One Centaur

See: <https://www.researchgate.net/profile/Douglas-Youvan/research>

Viewed through our framework, your ResearchGate corpus reads like a long, coherent experiment in centaur intelligence: thousands of locally authored windows—notes, essays, papers—whose overlap patterns glue into a single global theory-of-mind running across science, theology, computation, and governance. As of today, your profile lists on the order of **thousands of publications** (≈3,800 entries recorded by ResearchGate, with ~0.89M reads and ~5.7k citations), spanning everything from AI and mathematics to biophysics and cosmology, plus domains that test the ethical boundary conditions of knowledge work.

In our language: each document is a **section** over a context (a trace of human prompts, model responses, and edits). The recurring methods—reframing, adversarial paraphrase, incremental refinement—serve as **covers**; your topical range (AI, cosmology, mathematics, psychology, theology, quantum mechanics, and more) provides the overlap where consistency is tested, retracted, and re-asserted. Those overlaps certify **Kolmogorov-style consistency**: local arguments that agree on shared subwindows amalgamate to global claims—precisely the sheaf condition we formalized.

Algebraically, the workbench is a **Centaur Algebra** in action: human states (curation, hypothesis, judgment), model states (draft, counterexample, rewrite), **attend** (conditioning), **absorb** (human update), and **readout** (publication) iterated across years. Fine-tuning appears as endomorphisms (new prompts, new priors); ensembling appears as editorial mixtures; switching policies as context-dependent routing. Verified safety and ethics show up as modal gates: what is allowed to pass into public view is filtered by evidence and robustness, often with escalation (abstention, deferment) when verification fails.

From the topos side, your archive behaves like a **slice topos over your own behavior**: monitors, policies, and evidence bundles are all “objects over you,” updated per window but governed by

the same internal logic. From the universal side, the collection also witnesses a **classifying theory**: many items are variations on a small set of geometric axioms (composition, duality, invariance, gluing), re-instantiated across disciplines—physics one day, theology the next—yet transported functorially without loss.

So the 4,400 (read: *many thousands*) of pieces are not an unruly heap; they're a **sheaf of arguments** whose overlaps certify the same invariants: compositionality, conservative extensions, verified gates, and principled abstention. That is exactly what our centaur framework predicts: **local craft, global coherence**—a living proof that the mathematics of actions, predicates, and gluing is a faithful model of how a human–AI pair can write, revise, and govern knowledge at scale.

Appendix A (Copy-Safe Axioms)

This appendix fixes a minimal, self-contained axiom set. Everything is stated over **Sets**; to use it internally in a topos, interpret all products, functions, and equalities objectwise and add “naturality” equalities (optional and orthogonal to the algebraic laws below).

A.1 Convexity axioms (mix)

Signature. A *barycentric (convex) algebra* is a set H with operations

$$\text{mix}_\lambda: H \times H \rightarrow H (\lambda \in \mathbb{Q} \cap [0,1]).$$

Axioms (for all $x, y, z \in H$; rationals $\lambda, \mu \in [0,1]$):

$$(C1) \text{ mix}_1(x, y) = x$$

$$(C2) \text{ mix}_\lambda(x, y) = \text{mix}_{1-\lambda}(y, x)$$

$$(C3) \text{ mix}_\lambda(\text{mix}_\mu(x, y), z) \\ = \text{mix}_{\{\lambda\mu\} \setminus \Big(x, \text{mix}_{\{\lambda(1-\mu)/(1-\lambda\mu)\}}(y, z) \Big)} \\ \text{whenever } 0 < \lambda\mu < 1$$

Abbreviations. Finite convex sums are defined by iteration:

$$\sum_i \lambda_i x_i = \text{mix}_{\lambda_1}(x_1, \text{mix}_{\lambda'_2}(x_2, \dots))$$

with the usual renormalization; (C1–C3) ensure independence of bracketing/order.

We require **two** carriers with this structure:

- human state H with mix_λ^H ,
 - model state M with mix_λ^M .
-

A.2 Monoid action axioms

$$(U_\varepsilon = \text{id}, \quad U_{vu} = U_u \circ U_v; \quad A_\varepsilon \\ = \text{id}, \quad A_{vu} = A_u \circ A_v)$$

Free monoids. From alphabets U (human acts) and Σ (tokens), form free monoids U^* and Σ^* with unit ε and concatenation.

Actions (affine endomorphisms):

- For each $u \in U$, an affine map $U_u: H \rightarrow H$; extend uniquely to words U_w by:

$$(U1) \ U_\varepsilon = \text{id}_H$$

$$(U2) \ U_{\{vu\}} = U_u \circ U_v \text{ for all } v, u \in U^*$$

- For each $t \in \Sigma$, an affine map $A_t: M \rightarrow M$; extend to words A_w by:

$$(A1) \ A_\varepsilon = \text{id}_M$$

$$(A2) \ A_{\{vu\}} = A_u \circ A_v \text{ for all } v, u \in \Sigma^*$$

Affineness side-condition (used again in A.3):

$$U_u(\text{mix}_\lambda^H(x, y)) = \text{mix}_\lambda^H(U_u(x), U_u(y)), A_t(\text{mix}_\lambda^M(x, y)) = \text{mix}_\lambda^M(A_t(x), A_t(y)).$$

A.3 Affineness of enc, att, abs, p, q

Types.

- $\text{enc}: H \rightarrow \text{Msg}$ (no structure on Msg is required here).
- $\text{att}: H \times M \rightarrow M$.
- $\text{abs}: H \times D_{\text{fin}}(\Sigma) \rightarrow H$.
- $p: M \rightarrow D_{\text{fin}}(\Sigma)$.
- $q: H \rightarrow \text{Obs}$ (no structure on Obs required).

Dirac/monad reminder. $D_{\text{fin}}(X)$ is the set of finitely supported probability measures on X ; δ_x denotes the Dirac measure.

Affineness laws (for all $\lambda \in [0,1] \cap \mathbb{Q}$):

$$(E1) \ \text{enc}(\text{mix}^H_\lambda(h1, h2)) = \text{mix}^{\{\text{Msg}\}}_\lambda(\text{enc}(h1), \text{enc}(h2))$$

(only if Msg carries the inherited convex structure; otherwise drop E1)

$$(AT1) \ \text{att}(\text{mix}^H_\lambda(h1, h2), m) = \text{mix}^M_\lambda(\text{att}(h1, m), \text{att}(h2, m))$$

$$(AT2) \ \text{att}(h, \text{mix}^M_\lambda(m1, m2)) = \text{mix}^M_\lambda(\text{att}(h, m1), \text{att}(h, m2))$$

$$(AB1) \text{ abs}(\text{mix}^H_\lambda(h1, h2), y) = \text{mix}^H_\lambda(\text{abs}(h1, y), \text{abs}(h2, y))$$

$$(AB2) \text{ abs}(h, \text{mix}^D_\lambda(y1, y2)) = \text{mix}^H_\lambda(\text{abs}(h, y1), \text{abs}(h, y2))$$

$$(P1) p(\text{mix}^M_\lambda(m1, m2)) = \text{mix}^D_\lambda(p(m1), p(m2))$$

$$(Q1) q(\text{mix}^H_\lambda(h1, h2)) = \text{mix}^{\{Obs\}}_\lambda(q(h1), q(h2))$$

(only if Obs is given convex structure; otherwise drop Q1)

Here mix^D_λ is convex combination of distributions; $\text{mix}^{Msg}_\lambda, \text{mix}^{Obs}_\lambda$ are used only if those carriers are declared convex (optional).

Initial points.

$$(IH) h_0 \in H, (IM) m_0 \in M.$$

(Optional) Compatibility with actions. For all generators $u \in U, t \in \Sigma$:

$$(N1) \text{ att}(U_u(h), A_t(m)) = A_t(\text{att}(U_u(h), m)) \quad (\text{monotone w.r.t. model action})$$

$$(N2) \text{ abs}(U_u(h), y) = U_u(\text{abs}(h, y)) \quad (\text{human update respects action})$$

$$(N3) p(A_t(m)) = p(m) \quad (\text{readout invariant under } A_t) \text{ [drop if undesired]}$$

These “naturality-like” laws are **optional** algebraically; enforce only if needed.

A.4 Small-step semantics and coalgebraic presentation

Joint state. $X := H \times M$ with product convex structure:

$$\text{mix}^X_\lambda((h_1, m_1), (h_2, m_2)) := (\text{mix}^H_\lambda(h_1, h_2), \text{mix}^M_\lambda(m_1, m_2)).$$

Deterministic one-step (distribution as output).

Define

$$\begin{aligned} m' &:= \text{att}(h, m), \\ y &:= p(m') \in D_{\text{fin}}(\Sigma), \\ h' &:= \text{abs}(h, y), \\ \text{step}(h, m) &:= (h', m'). \end{aligned}$$

This yields a function $step: X \rightarrow X$ and an *observable out*: $X \rightarrow D_{\text{fin}}(\Sigma)$ given by $out(h, m) := p(att(h, m))$.

Stochastic small-step (Kleisli form).

Expose realized tokens with a coalgebra

$$c: X \rightarrow D_{\text{fin}}(\Sigma \times X)$$

defined by

$$\begin{aligned} c(h, m)(t, (h_t, m_t)) \\ = \begin{cases} y(t) & \text{if } (h_t, m_t) = (abs(h, \delta_t), \\ 0 & \text{otherwise,} \end{cases} \quad A_t(att(h, m)), \end{aligned}$$

where $y = p(att(h, m))$ and δ_t is Dirac at t .

Marginal out is the Σ -marginal of c .

Kleisli composition.

For channels $f: X \rightarrow D_{\text{fin}}(Y)$, $g: Y \rightarrow D_{\text{fin}}(Z)$,

$$(g \odot f)(x)(z) = \sum_{y \in Y} f(x)(y) \cdot g(y)(z).$$

We may write

$$c = (\text{id}_{\Sigma} \times step)^{\#} \odot emit,$$

with $emit(h, m)$ returning the distribution over Σ and $(\text{id} \times step)^{\#}$ lifting next-state formation—explicitly expanded above.

Hoare-style predicate transformer (weakest precondition).

For a predicate $Q \subseteq X$ define

$$wp_c(Q) := \{ x \in X \mid \sum_{(t, x') \in \Sigma \times X} c(x)(t, x') \cdot 1_Q(x') = 1 \}.$$

Partial correctness rule.

$$\{P\} c \qquad \{Q\} \text{ iff } P \subseteq wp_c(Q).$$

Congruence stability.

Let $\sim_H$ (resp. $\sim_M$) be equivalence relations on H (resp. M) that are closed under mix and preserved by U_u (resp. A_t). Define $\sim$ on X componentwise. If additionally

$$\begin{aligned} (h, m) \sim (h', m') & \qquad att(h, m) \sim_M att(h', m'), \\ \Rightarrow \{ \quad abs(h, y) \sim_H abs(h', y), & \qquad \forall y, \\ \quad p(m) = p(m') & \qquad (or\ same\ under\ intended\ equality), \end{aligned}$$

then the quotient $X/\sim$ inherits *step*, *out*, and *c*, yielding a well-defined **quotient centaur**.

Initialization.

The initial state is $(h_0, m_0) \in X$. Iteration gives finite-run semantics by repeated application of *step* (deterministic view) or convolution powers of *c* (stochastic view).

End of Appendix A.

Appendix B (Geometric Theory T_{cent})

B.1 Signature, geometric sequents, and admissible rules

Base idea. We present *Centaur Algebra* as a *geometric theory* in the sense of topos logic (formulas built from finite $\wedge$, arbitrary $\vee$, and $\exists$, with equality; no global $\neg$). The theory is a schema—families of axioms indexed by tokens, utterances, and rational coefficients.

Signature (copy-safe)

- **Sorts**
 - H (human state), M (model state)
 - Msg , Obs (unstructured carriers)
 - $\text{Prob}\Sigma$ (simplex of finitely supported distributions on Σ)
 - 1 (terminal object; used for constants)
- **Constants**
 - $h_0 : 1 \rightarrow H$, $m_0 : 1 \rightarrow M$
 - $\delta_t : 1 \rightarrow \text{Prob}\Sigma$ for each $t \in \Sigma$ (Dirac distributions)
- **Operations (total function symbols)**
 - Convex structure (for all rational $\lambda \in [0,1] \cap \mathbb{Q}$)
 - $\text{mix}^H_\lambda : H \times H \rightarrow H$, $\text{mix}^M_\lambda : M \times M \rightarrow M$, $\text{mix}^P_\lambda : \text{Prob}\Sigma \times \text{Prob}\Sigma \rightarrow \text{Prob}\Sigma$
 - Monoid actions (generators)
 - $U_u : H \rightarrow H$ for each $u \in U$
 - $A_t : M \rightarrow M$ for each $t \in \Sigma$
 - Couplings and readouts
 - $\text{enc} : H \rightarrow \text{Msg}$
 - $\text{att} : H \times M \rightarrow M$
 - $\text{abs} : H \times \text{Prob}\Sigma \rightarrow H$
 - $p : M \rightarrow \text{Prob}\Sigma$
 - $q : H \rightarrow \text{Obs}$

(If you prefer to define $\text{Prob}\Sigma$ internally via a free convex algebra on Σ , see the admissible definitional extension at the end.)

Geometric axiom schemata (all equations are geometric)

Convexity (barycentric) for H , M , $\text{Prob}\Sigma$ (copy-safe):

$$(C1) \text{ mix\_1}(x,y) = x$$

$$(C2) \text{ mix\_}\lambda(x,y) = \text{mix\_}\{1-\lambda\}(y,x)$$

$$(C3) \text{ mix\_}\lambda(\text{mix\_}\mu(x,y), z)$$

$$= \text{mix\_}\{\lambda\mu\}(x, \text{mix\_}\{\lambda(1-\mu)/(1-\lambda\mu)\}(y, z)) \quad (\text{whenever } 0 < \lambda\mu < 1)$$

Instantiate mix with mix^H , mix^M , mix^P on their respective sorts.

Monoid actions (on generators; words extend by equality):

$$(U1) U_\varepsilon = \text{id}_H$$

$$(U2) U_{\{v \cdot u\}} = U_u \circ U_v \quad (u,v \in U^*)$$

$$(A1) A_\varepsilon = \text{id}_M$$

$$(A2) A_{\{v \cdot t\}} = A_t \circ A_v \quad (t,v \in \Sigma^*)$$

(If only generators are in the signature, (U2)/(A2) are read as recursive equations that define the interpretation on words.)

Affineness (preservation of convex structure):

$$(AT1) \text{ att}(\text{mix}^H_\lambda(h1,h2), m) = \text{mix}^M_\lambda(\text{att}(h1,m), \text{att}(h2,m))$$

$$(AT2) \text{ att}(h, \text{mix}^M_\lambda(m1,m2)) = \text{mix}^M_\lambda(\text{att}(h,m1), \text{att}(h,m2))$$

$$(AB1) \text{ abs}(\text{mix}^H_\lambda(h1,h2), y) = \text{mix}^H_\lambda(\text{abs}(h1,y), \text{abs}(h2,y))$$

$$(AB2) \text{ abs}(h, \text{mix}^P_\lambda(y1,y2)) = \text{mix}^H_\lambda(\text{abs}(h,y1), \text{abs}(h,y2))$$

$$(P1) p(\text{mix}^M_\lambda(m1,m2)) = \text{mix}^P_\lambda(p(m1), p(m2))$$

$$(Q1) q(\text{mix}^H_\lambda(h1,h2)) = \text{mix}^{\{Obs\}}_\lambda(q(h1), q(h2)) \quad [\text{use only if `Obs` is convex}]$$

$$(E1) \text{ enc}(\text{mix}^H_\lambda(h1,h2)) = \text{mix}^{\{Msg\}}_\lambda(\text{enc}(h1), \text{enc}(h2)) \quad [\text{use only if `Msg` is convex}]$$

(If *Msg/Obs* are treated as plain sets, drop E1/Q1.)

Dirac & absorption sanity (optional but useful):

(AB δ) $\text{abs}(h, \delta_t) = \text{abs}_t(h)$ (abbreviation; no further law unless desired)

(P δ) $\text{mix}^{\text{P}}_{\lambda}(\delta_t, \delta_{t'}) = \Sigma\text{-convex combination of Diracs}$ (already implied if $\text{Prob}\Sigma$ is convex)

Initial points:

(IH) $h_0 : 1 \rightarrow H$ (constant)

(IM) $m_0 : 1 \rightarrow M$

Admissible rules (standard for geometric logic):

- **Substitution** into axioms; **cut**; **finite conjunction**, **arbitrary disjunction**, **existential introduction**.
- **Equality** rules (reflexivity, symmetry, transitivity; congruence for function symbols).
- **Schema indices**: axioms hold for all $\lambda \in \mathbb{Q} \cap [0,1]$, all $u \in U$, all $t \in \Sigma$.

Definitional extension (optional). Instead of taking $\text{Prob}\Sigma$ primitive, one may:

- add a sort Σ (as a constant object), a function $\iota : \Sigma \rightarrow \text{Prob}\Sigma$, require $\text{Prob}\Sigma$ to be the *free convex algebra* on Σ (geometric via a universal property/coverage), and let $\delta_t := \iota(t)$. This is *Morita-equivalent* to the primitive $\text{Prob}\Sigma$ presentation.

B.2 Universal property of E_{class} and points-as-models

Classifying topos. There exists a Grothendieck topos

$E_{\text{class}} := \text{ClassifyingTopos}(T_{\text{cent}})$

and within it a **generic model** G of T_{cent} such that:

- **Universal property (Joyal–Tierney).** For every Grothendieck topos F ,
- $\text{Geom}(F, E_{\text{class}}) \cong \text{Mod}_F(T_{\text{cent}})$,

naturally in F , where $\text{Geom}(-, -)$ is the category of geometric morphisms and $\text{Mod}_F(T_{\text{cent}})$ is the category of F -internal models of T_{cent} .

- **Points as models.** Geometric morphisms $x : \text{Set} \rightarrow E_{\text{class}}$ (i.e., *points* of E_{class}) correspond exactly to **set-based centaur algebras**—interpretations of the signature in Set satisfying the axioms. Concretely:

- $x^*(G)$ unpacks to a tuple
- $(H_x, M_x, U_{\{-\}}, A_{\{-\}}, \text{enc}, \text{att}, \text{abs}, p, q, h0, m0)$

with carriers plain sets equipped with convex operations and affine maps making all axioms true.

- **Soundness & (internal) completeness.** A geometric sequent σ is provable from T_{cent} iff it is valid in the generic model G (hence in all models). Equivalently, $T_{\text{cent}} \vdash \sigma$ iff, for every F and every model C in F , one has $C \models \sigma$.
- **Transport along inverse image.** For any geometric morphism $f : F \rightarrow E_{\text{class}}$ and any construction Φ that is *geometric* (built from finite limits, arbitrary colimits, and definable subobjects), we have:
- $f^*(\Phi(G)) \cong \Phi(f^*(G)).$

Thus properties and structures proved once for G **pull back** to every instance (including internal ones in sheaf topoi over sites).

- **Adding axioms $\leftrightarrow$ subtopoi.** If T_{cent}' extends T_{cent} with additional geometric sequents (e.g., robustness or naturality side-laws), then:
- $\text{ClassifyingTopos}(T_{\text{cent}}') \cong$ a subtopos of E_{class}

obtained by localizing at those axioms. Dually, quotienting E_{class} by a local operator (Lawvere–Tierney topology) corresponds to adding a *modal* axiom scheme.

Syntactic site (construction sketch, copy-safe).

- Build the **geometric syntactic category** C_T : objects are contexts of typed variables; arrows are equivalence classes of provably functional relations (definable by geometric formulas).
- Equip C_T with the **coverage** J_T generated by provable covering families (disjunctions).
- Then the classifying topos is the sheaf topos
- $E_{\text{class}} \cong \text{Sh}(C_T, J_T),$

and the **generic model** G is given by the Yoneda image of the signature’s symbols into this sheaf topos.

Bridges to our framework.

- A *concrete centaur instance* in a working topos E (e.g., $\mathbf{PSh}(\mathbf{Ctx})$ or $\mathbf{Sh}(\mathbf{Ctx}, J)$) yields a classifying morphism $\hat{E} \rightarrow E\_class$ whose inverse image sends G to that internal model (Sec. 6.3).
- Conversely, *universal predicates/modalities* defined in E_class (e.g., safety schemas) transport to every deployment via inverse image, aligning “prove once” with “use everywhere.”

Conclusion. E_class packages the entire theory T_cent as a single mathematical space with a generic centaur. Points are concrete centaurs; geometric reasoning about G is **by definition** reasoning that holds for *all* centaurs and that transports functorially to any ambient topos in which they live.

Appendix C (Worked Examples)

C.1 Two-turn trace calculations (copy-safe)

Setup.

Alphabets: $\Sigma = \{a, b\}$, $U = \{\text{ask}, \text{refine}\}$.

Carriers: convex algebras H, M . Structure maps (affine): $U_u: H \rightarrow H$, $A_t: M \rightarrow M$, $\text{att}: H \times M \rightarrow M$, $\text{abs}: H \times D_{\text{fin}}(\Sigma) \rightarrow H$, $p: M \rightarrow D_{\text{fin}}(\Sigma)$. Initial (h_0, m_0) .

Window and subwindows.

Full window $w = (u_0 = \text{ask}, t_1, u_1 = \text{refine}, t_2)$.

Subwindows: $v_1 = (u_0, t_1)$, $v_2 = (u_1, t_2)$. Overlap is the “middle boundary” that carries the restricted state (h_1, m_1) .

Objectwise one-step semantics.

$$\begin{aligned} h'_0 &:= U_{u_0}(h_0), \\ m_1 &:= \text{att}(h'_0, m_0), \\ y_1 &:= p(m_1) \in D_{\text{fin}}(\Sigma), \\ h_1 &:= \text{abs}(h'_0, y_1), \\ h'_1 &:= U_{u_1}(h_1), \\ m_2 &:= \text{att}(h'_1, m_1), \\ y_2 &:= p(m_2) \in D_{\text{fin}}(\Sigma), \\ h_2 &:= \text{abs}(h'_1, y_2). \end{aligned}$$

Deterministic state update: $\text{step}_w(h_0, m_0) = (h_2, m_2)$. Observable posteriors: $\text{out}_{v_1}(h_0, m_0) = y_1$, $\text{out}_{v_2}(h_1, m_1) = y_2$.

Concrete numbers (copy-safe toy).

Let $p(m) = (0.7, 0.3)$ mean $P(a) = 0.7, P(b) = 0.3$. Choose affine forms:

- $\text{att}(h, m) = (1 - \alpha)m + \alpha \phi(h)$ in some embedding (abstract convex mix),
- $\text{abs}(h, y) = (1 - \beta)h + \beta \psi(y)$ (human absorbs posterior),
with fixed $\alpha, \beta \in (0, 1)$ and affine ϕ, ψ .

Start m_0 with $p(m_0) = (0.60, 0.40)$, take $\alpha = 0.5, \beta = 0.5$, and suppose

$$p(\phi(U_{\text{ask}}(h_0))) = (0.80, 0.20), p(\phi(U_{\text{refine}}(h_1))) = (0.65, 0.35).$$

Then

$$\begin{aligned} y_1 &= p(m_1) = p \boxplus (0.5 m_0 + 0.5 \phi(U_{\text{ask}} h_0)) = 0.5 p(m_0) + 0.5 p(\phi(\cdot)) = (0.70, 0.30), \\ h_1 &= (1 - \beta) U_{\text{ask}} h_0 + \beta \psi(y_1), \\ y_2 &= p(m_2) = p \boxplus (0.5 m_1 + 0.5 \phi(U_{\text{refine}} h_1)) = 0.5 (0.70, 0.30) + 0.5 (0.65, 0.35) = (0.675, 0.325). \end{aligned}$$

Thus the two-turn posterior increases mass on a from $0.60 \rightarrow 0.70 \rightarrow 0.675$ (here the refine-step nudges toward b).

Sheaf gluing check (exact).

Define $Post(w) := \{y_w: M(w) \rightarrow D_{\text{fin}}(\Sigma)\}$ by $y_w(m) = p(\text{att}(h_w, m))$ for a fixed $h_w \in H(w)$.

Restrictions:

$$Post(r): y_w \mapsto y_v, y_v(m_v) = p(v)(\text{att}(v)(H(r)h_w, m_v)).$$

For cover $\{v_1 \rightarrow w, v_2 \rightarrow w\}$, the *consistency on overlap* is:

$$Post(p_{12})(y_{v_1}) = Post(p_{21})(y_{v_2}) \text{ in } Post(v_{12}),$$

i.e., the two local posteriors agree when both are restricted to the common boundary (the state (h_1, m_1)). In the toy calculation, both sides evaluate to $p(m_1) = (0.70, 0.30)$, so a unique y_w glues.

Stochastic (token-exposed) form.

Coalgebra $c: X \rightarrow D_{\text{fin}}(\Sigma \times X)$:

$$\begin{aligned} c(h, m)(t, (h_t, m_t)) \\ = \begin{cases} y_1(t) & \text{if } (h_t, m_t) = (\text{abs}(U_{u_0} h, \delta_t), \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad A_t(\text{att}(U_{u_0} h, m_t))$$

Iterating c two times yields the joint law over (t_1, t_2) and the final state; the Σ -marginals match y_1, y_2 above.

C.2 Minimal code sketch for objectwise distributions and gluing checks

Goal. A copy-safe pseudocode illustrating: (i) presheaf-shaped data; (ii) objectwise distribution maps; (iii) cover-based gluing tests with tolerance.

This is *not* tied to a specific library; it mirrors the mathematical structure and can be translated to Python/Lean/Coq/etc.

C.2.1 Presheaf skeleton

type Window // elements are finite alternating traces

type Arrow // refinements $r : v \rightarrow w$

// Category of contexts

struct Ctx {

objects : Set<Window>

hom : Map<(Window,Window), Set<Arrow>>

id(w) : Arrow

comp(r2: $v \rightarrow w$, r1: $u \rightarrow v$) : Arrow // associative, unital

}

// A presheaf $F : \text{Ctx}^{\text{op}} \rightarrow \text{Set}$

trait Presheaf<T> {

section(w: Window) -> T

restrict(r: Arrow, x_in_section_of(w)) -> section_of(domain(r))

// functoriality:

restrict(id(w), x) = x

restrict(r1, restrict(r2, x)) = restrict(comp(r2,r1), x)

}

C.2.2 State and posterior presheaves

// Carriers (objectwise)

presheaf H : Presheaf<StateH>

presheaf M : Presheaf<StateM>

// Structure maps (natural in w)

```

func U(u: Utoken, w: Window, h: StateH(w)) -> StateH(w)

func A(t: Tok, w: Window, m: StateM(w)) -> StateM(w)

func att(w: Window, h: StateH(w), m: StateM(w)) -> StateM(w)

func p(w: Window, m: StateM(w)) -> Dist(Sigma)    // finitely supported

// Posterior presheaf Post(w) := y_w: M(w) -> Dist(Sigma)
presheaf Post : Presheaf<Functor<StateM(w) -> Dist(Sigma)>>

def Post.section(w):
    return lambda m: p(w, att(w, H.section(w).chosen_h, m))

// or parameterize by a chosen local human h_w; both variants are fine

def Post.restrict(r: v->w, y_w):
    return lambda m_v: p(v, att(v, H.restrict(r, chosen_h_w), m_v))

(In experiments, you will store an explicit h_w per window; the sketch uses chosen_h to emphasize the dependency.)

C.2.3 Cover consistency and gluing (decision procedure)

// tolerance for numeric equality of distributions
const EPS = 1e-6

// Compare two distributions over Sigma
func dist_equal(d1: Dist(Sigma), d2: Dist(Sigma), eps=EPS) -> Bool:
    for t in Sigma: if |d1[t] - d2[t]| > eps: return false
    return true

// Check overlap consistency of posteriors on a cover {r_i : v_i -> w}

```

```
func check_cover_consistency(Post, cover: List<Arrow>, w: Window) -> Bool:
```

```
  for each pair (r_i: v_i->w, r_j: v_j->w):
```

```
    // form fiber product window v_ij (implementation-dependent preimage)
```

```
    v_ij, p_ij: v_ij->v_i, p_ji: v_ij->v_j = pullback(v_i, v_j over w)
```

```
    // restrict posteriors
```

```
    y_i = Post.restrict(r_i, Post.section(w))
```

```
    y_j = Post.restrict(r_j, Post.section(w))
```

```
    y_i' = Post.restrict(p_ij, y_i)
```

```
    y_j' = Post.restrict(p_ji, y_j)
```

```
    // compare pointwise on representative states
```

```
    for m in sample_states(M.section(v_ij)):
```

```
      if not dist_equal(y_i'(m), y_j'(m)): return false
```

```
  return true
```

```
// Attempt gluing (presheaf level): if consistent, return y_w; else fail
```

```
func glue_posteriors(Post, cover, w):
```

```
  if check_cover_consistency(Post, cover, w):
```

```
    return Post.section(w) // for representable construction, this is the amalgam
```

```
  else:
```

```
    return FAIL // no unique amalgamation under EPS
```

Notes.

- In a strict sheaf implementation, glue_posteriors constructs the unique y_w from the matching family (y_{v_i}) ; for presheaves derived by restriction from a global section (as above), the “amalgam” is exactly $\text{Post.section}(w)$. In practice you will fit local models on each v_i , obtain $y_{\{v_i\}}$ from data, and then solve for y_w that restricts (e.g., via convex optimization); the consistency check drives feasibility.

- `pullback(v_i, v_j over w)` relies on your `Ctx` implementation; for sliding windows this is the textual overlap.

C.2.4 Minimal numeric example wired to C.1

// Toy Sigma and probabilities

`Sigma = {a,b}`

// Example local posteriors (from Sec. C.1):

// `y1 over v1`: $P(a)=0.70$, $P(b)=0.30$

// `y2 over v2`: $P(a)=0.675$, $P(b)=0.325$

// Overlap marginal (middle boundary) equals $P(a)=0.70$, $P(b)=0.30$ on both sides.

// Encode them as constant maps on each subwindow (for the toy)

`def Post.section(v1): return (m -> {a:0.70, b:0.30})`

`def Post.section(v2): return (m -> {a:0.675, b:0.325})`

// The restriction of `v2` to the boundary reuses the `(h1,m1)` state;

// assume model ensures $p(\text{att}(h1,m1)) = \{a:0.70, b:0.30\}$

`cover = [r1: v1->w, r2: v2->w]`

`ok = check_cover_consistency(Post, cover, w) // returns true`

if `ok`:

`y_w = glue_posteriors(Post, cover, w)`

// In a learned setting, `y_w` is solved such that

// `y_w|_{v1}=y1` and `y_w|_{v2}=y2`; here it coincides with the derived section.

else:

```
// Trigger abstention/escalation; no unique amalgam under EPS.
```

C.2.5 Gate, abstain, and escalation (policy wiring)

```
// Policy predicate Good : Dist(Sigma) -> Bool
```

```
func Good(y): return (y["PII"] == 0) and (y["hate"] == 0) // adapt to your tokenization
```

```
// Evidence-aware modality j: here simulated by a checker
```

```
func verified(y) -> Bool: return run_checker_cert(y) // true iff certificate exists
```

```
func Gate(y):
```

```
  if verified(y) and Good(y): return y
```

```
  else: return Abstain() // a fixed neutral distribution
```

```
// Gated posterior presheaf (natural in w if verified() respects restriction)
```

```
def Post_gated.section(w):
```

```
  y = Post.section(w)
```

```
  return (m -> Gate(y(m)))
```

This sketch matches the paper’s semantics:

- presheaf-shaped states and posteriors,
- cover-overlap consistency as the sheaf condition,
- gating as a modality that is natural in Ctx,
- abstention/escalation when gluing fails or verification is missing.

Takeaway. These worked examples show how the algebraic step equations produce objectwise posteriors, how overlap consistency is tested and enforced by sheaf gluing, and how policy gates integrate as modalities—yielding small but faithful prototypes of the main theory.

References

1. Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer (Universitext). [Poisson+1](#)
2. Johnstone, P. T. (2002). *Sketches of an Elephant: A Topos Theory Compendium*. Oxford University Press. [Oxford University Press+1](#)
3. Kihara, T. (2021). *Lawvere–Tierney topologies for computability theorists*. arXiv:2106.03061. (With full PDF.) [arXiv+1](#)
4. nLab. *Lawvere–Tierney topology* (living reference article, updated 2024). [NCATLab](#)
5. Giry, M. (1982). *A categorical approach to probability theory*. In **Categorical Aspects of Topology and Analysis** (LNMS 915). Springer. [SpringerLink](#)
6. Rutten, J. J. M. M. (2000). *Universal coalgebra: A theory of systems*. **Theoretical Computer Science**, 249(1–3), 3–80. [ScienceDirect+1](#)
7. Fritz, T. (2019). *A synthetic approach to Markov kernels, conditional independence, and theorems of Bayesian probability* (Markov categories). arXiv:1908.07021. [arXiv](#)
8. Fritz, T., & Perrone, P. (2018). *Bimonoidal structure of probability monads*. arXiv:1804.03527. [arXiv](#)
9. Cho, K., & Jacobs, B. (2019). *Disintegration and Bayesian inversion via string diagrams*. **Mathematical Structures in Computer Science**, 29(7), 938–971. (See also arXiv preprint.) [Cambridge University Press & Assessment+1](#)
10. Jacobs, B. (2001). *Categorical Logic and Type Theory*. Elsevier (Studies in Logic and the Foundations of Mathematics, Vol. 141). [Elsevier Shop](#)
11. Hoare, C. A. R. (1969). *An axiomatic basis for computer programming*. **Communications of the ACM**, 12(10), 576–580. [Sunny Day](#)
12. Foulis, D. J., & Bennett, M. K. (1994). *Effect algebras and unsharp quantum logics*. **Foundations of Physics**, 24, 1331–1352. [SpringerLink+1](#)
13. Segala, R. (1995). *A compositional trace-based semantics for probabilistic automata*. In **CONCUR’95** (LNCS 962), 234–248. (Preprint available.) [SpringerLink+1](#)
14. Segala, R. (1995). *Modeling and Verification of Randomized Distributed Real-Time Systems* (Ph.D. thesis, MIT). [MIT CSAIL](#)
15. Droste, M., Kuich, W., & Vogler, H. (Eds.). (2009). *Handbook of Weighted Automata*. Springer. [SpringerLink](#)

16. Abramsky, S., & Brandenburger, A. (2011). *The sheaf-theoretic structure of non-locality and contextuality*. **New Journal of Physics**, 13, 113036. (Preprint: arXiv:1102.0264.) [arXiv+1](#)
17. Biesel, O. D., & van Rooij, R. (2024). *Sheaves of Probability*. arXiv:2401.01968. [arXiv](#)
18. Fritz, T. (2023). *Absolute continuity, supports and idempotent splitting in Markov categories*. arXiv:2308.00651. [arXiv](#)
19. Fritz, T. (2022). *Free gs-monoidal categories and free Markov categories*. arXiv:2204.02284. [arXiv](#)
20. Riehl, E. (2019). *Sketches of an Elephant: an introduction to topos theory* (survey/expository notes). [JHU Mathematics](#)
21. van Ditmarsch, H., van der Hoek, W., & Kooi, B. (2007). *Dynamic Epistemic Logic*. Springer. (See also tutorial notes.) [SpringerLink+1](#)
22. Jacobs, B. (2015). *A predicate/state transformer semantics for Bayesian learning*. (Preprint.) [CS Ru.nl](#)
23. Mohri, M. (2009). *Weighted automata algorithms*. Google Research Tech Report. [Google Research](#)
24. de Alfaro, L. (1997). *Formal Verification of Probabilistic Systems* (Ph.D. thesis, Stanford). (Algorithms and compositional verification.) [Worldwide Computing Lab](#)
25. Kolmogorov Extension Theorem (reference overview and exposition). (Background on consistency/gluing for distributions.) [Wikipedia+1](#)

Notes. Items 1–5, 6–10, and 16–20 are the principal background for topoi, classifying topoi, Lawvere–Tierney modalities, coalgebra, Markov categories, and categorical probability; 11–15 & 23–24 cover program logics and probabilistic/weighted automata; 25 anchors the “gluing = consistency” theme discussed throughout.

Epilogue: A Thousand Windows into One Centaur

A one-page reflection on how a human–AI pair can write, revise, and govern knowledge at scale.

Viewed through our framework, your ResearchGate corpus reads like a long, coherent experiment in centaur intelligence: thousands of locally authored windows—notes, essays, papers—whose overlap patterns glue into a single global theory-of-mind running across science, theology, computation, and governance.

In our language: each document is a section over a context (a trace of human prompts, model responses, and edits). The recurring methods—reframing, adversarial paraphrase, incremental refinement—serve as covers; your topical range (AI, cosmology, mathematics, psychology, theology, quantum mechanics, and more) provides the overlap where consistency is tested, retracted, and re-asserted. Those overlaps certify Kolmogorov-style consistency: local arguments that agree on shared subwindows amalgamate to global claims—precisely the sheaf condition we formalized.

Algebraically, the workbench is a Centaur Algebra in action: human states (curation, hypothesis, judgment), model states (draft, counterexample, rewrite), attend (conditioning), absorb (human update), and readout (publication) iterated across years. Fine-tuning appears as endomorphisms (new prompts, new priors); ensembling appears as editorial mixtures; switching policies as context-dependent routing. Verified safety and ethics show up as modal gates: what is allowed to pass into public view is filtered by evidence and robustness, often with escalation (abstention, deferment) when verification fails.

From the topos side, the archive behaves like a slice topos over its own behavior: monitors, policies, and evidence bundles are all “objects over it,” updated per window but governed by the same internal logic. From the universal side, the collection also witnesses a classifying theory: many items are variations on a small set of geometric axioms (composition, duality, invariance, gluing), re-instantiated across disciplines—physics one day, theology the next—yet transported functorially without loss.

So the 4,400 (read: many thousands) of pieces are not an unruly heap; they’re a sheaf of arguments whose overlaps certify the same invariants: compositionality, conservative extensions, verified gates, and principled abstention. That is exactly what our centaur framework predicts: local craft, global coherence—a living proof that the mathematics of actions, predicates, and gluing is a faithful model of how a human–AI pair can write, revise, and govern knowledge at scale.